



Universidad  
Carlos III de Madrid

Departamento de Ingeniería Telemática

PROYECTO DE FIN DE CARRERA

# Extensiones y mejoras de la herramienta CourseEditor

Autor: Raquel Catalina Vaquero

Tutor: Pedro José Muñoz Merino

Director: M<sup>a</sup> Carmen Fernández Panadero

Leganés, julio de 2013



Título: Extensiones y mejoras de la herramienta CourseEditor

Autor: Raquel Catalina Vaquero

Tutor: Pedro José Muñoz Merino

Director: M<sup>a</sup> Carmen Fernández Panadero

## EL TRIBUNAL

Presidente: Isaac Seoane Pujol

Vocal: Jesús Cid-Sueiro

Secretario: Derrick Leony

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 11 de Julio de 2013 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE



# Agradecimientos

Me gustaría utilizar estas líneas para expresar mi más sincero agradecimiento a todas las personas que han confiado en mí y que sin su apoyo no me hubiera sido posible finalizar este proyecto.

Gracias a toda mi familia, especialmente a los que sé que una vez conseguida esta meta, estéis orgullosos de mí, tanto aquí, como allá arriba.

Gracias a mi hermana, por su infinito apoyo, consejos y ayuda. Porque como ella dice, "Nadie dijo que iba a ser fácil" y porque, ahora sí, esto es el comienzo de lo mejor que está por venir y siempre a su lado.

Finalmente, gracias a los tutores de este proyecto, Carmen Fernández Panadero y Pedro J. Muñoz Merino, por su dedicación y ánimo durante todo el transcurso del mismo.

Este proyecto de fin de carrera supone el final de una etapa que no hubiese podido completar sin la ayuda, el apoyo y los consejos de muchas personas. A todos ellos, gracias.



# Resumen

El objetivo principal de este proyecto de fin de carrera es crear una nueva versión de la herramienta CourseEditor desarrollada en un proyecto de fin de carrera anterior, con el objetivo de mejorar tanto la usabilidad como la funcionalidad de la misma.

CourseEditor es una herramienta software implementada en Java, que permite a un profesor describir la programación curricular de un curso pasando por sus diferentes fases (objetivos, contenidos, secuenciación y evaluación), generando ficheros que son compatibles con un subconjunto de la especificación de e-learning IMS-LD (Learning Design) y otros ficheros con información adicional que no está recogida en IMS-LD.

Las mejoras respecto a la usabilidad permiten al profesor crear cursos de una forma más fácil e intuitiva mediante la mejora de la interfaz y solución de errores existentes en la versión anterior. Por otra parte las mejoras funcionales de la herramienta se han centrado en dos puntos, la introducción de características del nivel B recogidas en la especificación IMS Learning Design y la creación de un editor gráfico, ambas orientadas a facilitar la secuenciación de tareas.

Finalmente se ha realizado una evaluación de la herramienta mediante la realización de pruebas funcionales y estructurales así como la realización de diferentes secuenciaciones con el objetivo de verificar que la versión final es estable.

El resultado del trabajo llevado a cabo es una herramienta más completa, fácil de usar e intuitiva.

**Palabras Clave:** *E-learning, IMS-LD, CourseEditor.*





# Abstract

The main objective of this final degree project is the development of a new version of the existing tool called CourseEditor which was developed in other final degree project. Its main aim is to improve the usability of the tool and also to add more functionality to it.

CourseEditor is a software tool implemented in Java. It allows teachers/lecturers describe the complete planning of a course through its different phases (objectives, contents, sequencing and evaluation) as it is detailed in the IMS Learning Design specification. At the same time, this tool is capable of generating files according to the so called specification describing the methodology used. Moreover, the tool is able to generate other XML files with information related to other planning topics of a course that are not included in IMS-LD.

Usability improvements allow lecturers to generate courses in an easier and more intuitive way, thanks to the improvements that has been made on the tool's interface and the solution of errors that existed in the previous one. On the other hand, functionality improvements follow two main points, the introduction of level B functionalities that are included in the specification and the creation of a graphical editor, both of them oriented to facilitate the tasks' sequence.

Finally, an evaluation of the tool has been made by doing functional and structural tests, as well as designing different sequencings to prove that the final version of the tool is a stable one.

The final result is a more complete, intuitive and easy-to-use tool.

**Keywords:** *E-learning, IMS-LD, CourseEditor.*

# Índice general

<b>1. INTRODUCCIÓN .....</b>	<b>19</b>
1.1 Introducción .....	19
1.2 Objetivos .....	20
1.3 Estructura de la memoria.....	22
<b>2. ESTADO DEL ARTE .....</b>	<b>25</b>
2.1 Diseño curricular .....	25
2.1.1 Aportaciones del diseño curricular al proyecto .....	26
2.2 E-Learning.....	27
2.2.1 Introducción .....	27
2.2.2 Elementos de E-Learning .....	28
2.2.3 Ventajas e Inconvenientes del E-Learning .....	30
2.2.4 Aportaciones de E-Learning al proyecto.....	31
2.3 Especificaciones de E-Learning .....	32
2.3.1 Introducción .....	32
2.3.2 Institute for Electrical and Electronic Engineers Learning Technology Standards Committee (IEEE LTSC) .....	33
2.3.3 Advanced Distributed Learning (ADL) .....	33
2.3.4 Aviation Industry CBT Committee (AICC).....	34
2.3.5 IMS Global Learning Consortium.....	35
2.3 IMS-LD Learning Design .....	37
2.3.1 Introducción al estándar IMS-LD .....	37
2.3.2 Niveles de IMS LD.....	38
2.3.3 Principales limitaciones de la especificación.....	41
2.4 Herramientas que utilizan la especificación IMS-LD .....	42
2.4.1 Herramientas de autoría .....	42
2.4.2 Motores de ejecución.....	50
2.4.3 Reproductores.....	50
2.5 Tecnologías utilizadas .....	54
2.5.1 Java.....	54
2.5.2 XML.....	56
2.5.3 Eclipse .....	58

2.5.4	Bugzilla .....	58
<b>3.</b>	<b>DESARROLLO DEL PROYECTO .....</b>	<b>61</b>
3.1	Introducción .....	61
3.2	Cambios en la interfaz de la herramienta .....	63
3.2.1	Introducción de un Menú .....	63
3.2.2	Acceso a la ayuda de la herramienta .....	65
3.2.3	Selección de la ubicación de guardado de ficheros .....	67
3.2.4	Guardado automático de la información .....	68
3.2.5	Detección de errores producidos en la introducción de la información de un curso 70	
3.2.6	Cambios en la pestaña ‘Duración’ .....	72
3.2.7	Cambios en la pestaña ‘Objetivos’ .....	74
3.2.8	Cambios en la pestaña ‘U.D’ .....	78
3.2.9	Cambios en la pestaña ‘T.Tareas’ .....	81
3.2.10	Cambios en la pestaña ‘Tareas’ .....	83
3.2.11	Cambios en la pestaña ‘Ud-Tareas’ .....	94
3.2.12	Cambios en el Editor Gráfico .....	96
3.3	Errores en la herramienta .....	104
3.3.1	Problemas de aparición de información en los jcomboBox .....	104
3.3.2	Desaparición de roles al definir nuevas tareas .....	104
3.3.3	Nueva pestaña: ‘Roles’ .....	104
3.3.4	Eliminación de la necesidad de definir tareas soporte en un curso .....	107
3.4	Nuevas funcionalidades .....	107
3.4.1	Introducción de funcionalidad de nivel B .....	107
3.4.2	Persistencia de la secuenciación gráfica en el editor .....	112
<b>4.</b>	<b>IMPLEMENTACIÓN DEL PROYECTO .....</b>	<b>113</b>
4.1	Diseño e implementación .....	113
4.1.1	Clase PantallaInicial .....	115
4.1.2	Clases ModeloTablaObjetivos, ModeloTablaRoles, ModeloTablaTareas y ModeloTablaTipoTareas .....	117
4.1.3	Clase InfTareaPorSerTarea .....	118
4.1.4	Clase EditorGráfico .....	118
4.1.5	Clase MensajeSalir .....	121
4.1.6	Clases MensajeErrores VectorErrores .....	121
<b>5.</b>	<b>EVALUACIÓN DE LA HERRAMIENTA. ....</b>	<b>123</b>
5.1	Pruebas funcionales .....	124
5.2	Pruebas estructurales .....	133
5.3	Secuenciaciones realizadas .....	142
<b>6.</b>	<b>GESTIÓN DEL PROYECTO .....</b>	<b>147</b>
6.1	Fases del desarrollo .....	147
6.1.1	Planificación .....	147
6.1.2	Desarrollo .....	148
6.1.3	Documentación .....	149
6.2	Planificación temporal .....	149
6.3	Presupuesto .....	151
<b>7.</b>	<b>CONCLUSIONES Y TRABAJO FUTURO .....</b>	<b>155</b>
7.1	Conclusiones .....	155
7.2	Trabajo futuro .....	157
<b>8.</b>	<b>TUTORIAL PARA LA REALIZACIÓN DE MODIFICACIONES EN LA HERRAMIENTA.....</b>	<b>159</b>
1.	Instalación de Java JDK .....	159
2.	Instalación de Eclipse .....	166
3.	Uso de la herramienta Eclipse .....	168

## ÍNDICE general

4. Código de la aplicación .....	171
5. Principales clases y métodos de la aplicación .....	180
<b>9. ENUNCIADO DEL CURSO DE PRUEBA CREADO PARA LA EVALUACIÓN DE LA HERRAMIENTA. ....</b>	<b>201</b>
<b>10. SECUENCIACIONES DEL INTERFAZ GRÁFICO PROBADAS. ....</b>	<b>211</b>
<b>11. GLOSARIO .....</b>	<b>222</b>
<b>12. REFERENCIAS.....</b>	<b>224</b>

# Índice de figuras

<i>Figura 1 Arquitectura de la especificación IMS</i> .....	35
<i>Figura 2 Interfaz de CopperAuthor</i> .....	43
<i>Figura 3 Interfaz de Reload Editor</i> .....	44
<i>Figura 4 Interfaz de CoSMoS</i> .....	44
<i>Figura 5 Interfaz de MOT+</i> .....	45
<i>Figura 6 Interfaz de ASK-LTD</i> .....	46
<i>Figura 7 Interfaz de Learning Activity Management System</i> .....	46
<i>Figura 8 Interfaz de Alfabet</i> .....	47
<i>Figura 9 Interfaz de CopperCore player</i> .....	51
<i>Figura 10 Interfaz de SLeD</i> .....	51
<i>Figura 11 Interfaz de RELOAD Player</i> .....	52
<i>Figura 12 Interfaz de GRAIL</i> .....	52
<i>Figura 13 Pestaña Info. General de la herramienta CourseEditor v1.0</i> .....	63
<i>Figura 14 Pestaña Info. General de la herramienta CourseEditor v2.0</i> .....	64
<i>Figura 15 Detalle de las funcionalidades del menú de CourseEditor v2.0</i> .....	64
<i>Figura 16 Ventana de ayuda de CourseEditor v2.0</i> .....	66
<i>Figura 17 Ayuda contextual de CourseEditor v2.0</i> .....	67
<i>Figura 18 Selección de la ubicación de guardado en CourseEditor v2.0</i> .....	68
<i>Figura 19 Ventana emergente que pregunta si se desea sobrescribir en CourseEditor v2.0</i> .....	68
<i>Figura 20 Guardado de la información de cada pestaña en CourseEditor v1.0</i> .....	69
<i>Figura 21 Guardado de la información de cada pestaña en CourseEditor v2.0</i> .....	70
<i>Figura 22 Mensaje de cierre de la aplicación en CourseEditor v2.0</i> .....	70
<i>Figura 23 Mensaje que lista los errores cometidos en la introducción de datos en CourseEditor v2.0</i> .....	71
<i>Figura 24 Señalización en la pestaña Info.General de los errores cometidos al introducir la información en CourseEditor v2.0</i> .....	72
<i>Figura 25 Señalización en la pestaña Duración de los errores cometidos al introducir la información en CourseEditor v2.0</i> .....	72

## ÍNDICE DE FIGURAS

<i>Figura 26 Pestaña Duración de CourseEditor v1.0 .....</i>	<i>73</i>
<i>Figura 27 Pestaña Duración de CourseEditor v2.0 .....</i>	<i>74</i>
<i>Figura 28 Calendario y campos de texto libres añadidos de CourseEditor v2.0.....</i>	<i>74</i>
<i>Figura 29 Pestaña Objetivos de CourseEditor v1.0 .....</i>	<i>76</i>
<i>Figura 30 Pestaña Objetivos de CourseEditor v2.0 .....</i>	<i>76</i>
<i>Figura 31 Botones Editar y Borrar de la tabla de CourseEditor v2.0 .....</i>	<i>77</i>
<i>Figura 32 Tool tip en la tabla Objetivos de CourseEditor v2.0 .....</i>	<i>78</i>
<i>Figura 33 Mensaje de error de CourseEditor v2.0.....</i>	<i>78</i>
<i>Figura 34 Pestaña U.D de CourseEditor v1.0.....</i>	<i>79</i>
<i>Figura 35 Pestaña U.D de CourseEditor v2.0.....</i>	<i>79</i>
<i>Figura 36 Tool tip del nombre de la unidad didáctica en la tabla U.D de CourseEditor v2.0.....</i>	<i>80</i>
<i>Figura 37 Tool tip de la descripción de la unidad didáctica en la tabla U.D de CourseEditor v2.0 .....</i>	<i>80</i>
<i>Figura 38 Tool tip de los objetivos con los que se relaciona cada unidad didáctica en la tabla U.D de CourseEditor v2.0.....</i>	<i>80</i>
<i>Figura 39 Pestaña T.Tareas de CourseEditor v1.0 .....</i>	<i>82</i>
<i>Figura 40 Pestaña T.Tareas de CourseEditor v2.0 .....</i>	<i>82</i>
<i>Figura 41 Implementación de tool tip para una mejor visualización.....</i>	<i>83</i>
<i>Figura 42 Pestaña Tareas de CourseEditor v1.0 .....</i>	<i>84</i>
<i>Figura 43 Pestaña Tareas CourseEditor v2.0 y ventana de introducción de información adicional de una tarea que aparecerá automáticamente cuando se pulsa el botón Añadir.....</i>	<i>85</i>
<i>Figura 44 Mensaje de error. Introducción de un valor no numérico en el campo Duración.....</i>	<i>86</i>
<i>Figura 45 Mensaje de error. Debe de haber al menos un rol definido .....</i>	<i>86</i>
<i>Figura 46 Mensaje de error. Debe haber definida al menos una unidad didáctica.....</i>	<i>86</i>
<i>Figura 47 Tool tip del nombre de la tarea en la tabla Tareas en CourseEditor v2.0 .....</i>	<i>87</i>
<i>Figura 48 Tool tip del tipo de la tarea en la tabla Tareas en CourseEditor v2.0 .....</i>	<i>87</i>
<i>Figura 49 Tool tip de la unidad didáctica de la tarea en la tabla Tareas en CourseEditor v2.0 .....</i>	<i>87</i>
<i>Figura 50 Tool tip del identificador de la tarea en la tabla Tareas en CourseEditor v2.0 .....</i>	<i>87</i>
<i>Figura 51 Tool tip del objetivo con el que se relaciona la tarea en la tabla Tareas en CourseEditor v2.0 .....</i>	<i>88</i>
<i>Figura 52 Pantalla ‘Por ser tarea’ de CourseEditor v1.0.....</i>	<i>88</i>
<i>Figura 53. Pantalla ‘Por ser de un tipo’ de CourseEditor v1.0 .....</i>	<i>89</i>
<i>Figura 54 Pantalla ‘Por ser evaluable’ de CourseEditor v1.0.....</i>	<i>89</i>
<i>Figura 55 Pantalla ‘Por ser entregable’ de CourseEditor v1.0 .....</i>	<i>90</i>
<i>Figura 56 Pantalla ‘Por ser tarea’ de CourseEditor v2.0.....</i>	<i>91</i>
<i>Figura 57 Pantalla ‘Por ser de un tipo’ de CourseEditor v2.0 .....</i>	<i>92</i>
<i>Figura 58 Ventana de información adicional de una tarea de CourseEditor v2.0 .....</i>	<i>92</i>
<i>Figura 59 Pantalla ‘Por ser evaluable’ de CourseEditor v2.0.....</i>	<i>93</i>
<i>Figura 60 Pantalla ‘Por ser entregable’ de CourseEditor v2.0 .....</i>	<i>93</i>
<i>Figura 61 Pestaña Ud-Tareas de CourseEditor v1.0 .....</i>	<i>95</i>
<i>Figura 62 Pestaña Ud-Tareas de CourseEditor v2.0 .....</i>	<i>95</i>
<i>Figura 63. Ordenación de las unidades didácticas de CourseEditor v2.0.....</i>	<i>96</i>
<i>Figura 64 Editor gráfico de CourseEditor v1.0.....</i>	<i>97</i>
<i>Figura 65 Editor gráfico de CourseEditor v2.0.....</i>	<i>98</i>

<i>Figura 66 Ayuda básica para un usuario sin experiencia previa en el uso del editor gráfico de CourseEditor v2.0 .....</i>	<i>99</i>
<i>Figura 67 Visualización detallada de los iconos para cada uno de los tipos de tarea. .</i>	<i>100</i>
<i>Figura 68 Ventana de definición de un nuevo tipo de tarea .....</i>	<i>101</i>
<i>Figura 69 Mensaje de error tras una introducción errónea de imagen para el tipo nuevo .....</i>	<i>101</i>
<i>Figura 70 Icono por defecto para los nuevos tipos de tarea definidos por el usuario en CourseEditor v2.0 .....</i>	<i>102</i>
<i>Figura 71 Representación gráfica de CourseEditor v1.0 .....</i>	<i>103</i>
<i>Figura 72 Representación gráfica de CourseEditor v2.0 .....</i>	<i>103</i>
<i>Figura 73 Nueva pestaña para la definición de Roles de CourseEditor v2.0 .....</i>	<i>105</i>
<i>Figura 74 Campos que se han de rellenar para definir un nuevo rol de CourseEditor v2.0 .....</i>	<i>106</i>
<i>Figura 75 Ventana para la creación de nuevas etiquetas de rol de CourseEditor v2.0.</i>	<i>106</i>
<i>Figura 76 Tool tip de la descripción del rol de CourseEditor v2.0.....</i>	<i>107</i>
<i>Figura 77 Definición de la condición si la tarea es evaluable y entregable de CourseEditor v2.0 .....</i>	<i>109</i>
<i>Figura 78 Definición de la condición si la tarea es evaluable de CourseEditor v2.0.....</i>	<i>110</i>
<i>Figura 79 Definición de la condición si la tarea es entregable de CourseEditor v2.0 ...</i>	<i>111</i>
<i>Figura 80 Diagrama UML de la aplicación .....</i>	<i>114</i>
<i>Figura 81 Resultado de la prueba unitaria de la pestaña Info. General.....</i>	<i>134</i>
<i>Figura 82 Resultado de las pruebas unitarias de la pestaña Duración .....</i>	<i>136</i>
<i>Figura 83 Resultado de las pruebas unitarias de la pestaña Objetivos .....</i>	<i>138</i>
<i>Figura 84 Resultado de las pruebas unitarias de la pestaña U.D.....</i>	<i>139</i>
<i>Figura 85 Resultado de las pruebas unitarias de la pestaña T.Tareas .....</i>	<i>140</i>
<i>Figura 86 Resultado de las pruebas unitarias de la pestaña Roles.....</i>	<i>141</i>
<i>Figura 87 Resultado de las pruebas unitarias de la pestaña Tareas .....</i>	<i>142</i>
<i>Figura 88 Ejecución del script CopperCore.bat.....</i>	<i>143</i>
<i>Figura 89 Interfaz de CopperCore player. ....</i>	<i>143</i>
<i>Figura 90 Mensaje obtenido tras una validación con éxito. ....</i>	<i>144</i>
<i>Figura 91 Pasos realizados para la ejecución del Clicc .....</i>	<i>145</i>
<i>Figura 92 Interfaz de CopperCore player .....</i>	<i>146</i>
<i>Figura 93 Visualización de la ejecución de la unidad de aprendizaje en el reproductor. ....</i>	<i>146</i>
<i>Figura 94 Diagrama de Gantt .....</i>	<i>150</i>
<i>Figura 95. Asistente de instalación de Java Development kit .....</i>	<i>160</i>
<i>Figura 96. Asistente de instalación de Java Development kit .....</i>	<i>160</i>
<i>Figura 97 Asistente de instalación de Java Development kit .....</i>	<i>161</i>
<i>Figura 98 Asistente de instalación de Java Development kit.....</i>	<i>162</i>
<i>Figura 99 Asistente de instalación de Java Development kit .....</i>	<i>162</i>
<i>Figura 100 Asistente de instalación de Java Development kit .....</i>	<i>163</i>
<i>Figura 101 Ventana de propiedades del sistema .....</i>	<i>164</i>
<i>Figura 102 Ventana variables de entorno .....</i>	<i>165</i>
<i>Figura 103 Ventana para editar las variables de entorno del sistema.....</i>	<i>165</i>
<i>Figura 104 Ventana para añadir nueva variable de entorno al sistema.....</i>	<i>166</i>
<i>Figura 105 Consola de comandos de Windows 7 .....</i>	<i>166</i>
<i>Figura 106 Descarga de Eclipse.....</i>	<i>167</i>
<i>Figura 107 Ejecutable de Eclipse .....</i>	<i>168</i>
<i>Figura 108 Selección de workspace en Eclipse .....</i>	<i>168</i>
<i>Figura 109 Asistente de creación de un nuevo proyecto .....</i>	<i>169</i>

## ÍNDICE DE FIGURAS

<i>Figura 110 Ventana para importar proyectos existentes a Eclipse</i> .....	170
<i>Figura 111 Ventana para exportar archivos desde Eclipse</i> .....	171
<i>Figura 112 Opciones de la pestaña ‘Help’ del menú de Eclipse</i> .....	172
<i>Figura 113 Asistente de instalación de nuevo software en Eclipse</i> .....	173
<i>Figura 114 Añadir repositorio</i> .....	173
<i>Figura 115 Asistente de instalación de nuevo software en Eclipse</i> .....	174
<i>Figura 116 Asistente de instalación de nuevo software en Eclipse</i> .....	174
<i>Figura 117 Asistente de instalación de nuevo software en Eclipse</i> .....	175
<i>Figura 118 Ventana para abrir distintas perspectivas en Eclipse</i> .....	176
<i>Figura 119 Creación de un nuevo repositorio</i> .....	177
<i>Figura 120 Añadir nuevo repositorio</i> .....	178
<i>Figura 121 Selección de la opción ‘Update to version’</i> .....	179
<i>Figura 122 Selección de la opción Commit</i> .....	180
<i>Figura 123 Pantalla Inicial de CourseEditor</i> .....	181
<i>Figura 124 Ventana para crear nuevos tipos de tareas</i> .....	184
<i>Figura 125 Ventana para definir la plantilla de los nuevos tipos de tarea</i> .....	185
<i>Figura 126 Ventana para establecer una condición en la secuenciación de una unidad didáctica</i> .....	186
<i>Figura 127 Ventana para editar una tarea en el editor gráfico</i> .....	186
<i>Figura 128 Ventana para editar una caja de tipo Unión en el editor gráfico</i> .....	187
<i>Figura 129 Editor gráfico</i> .....	188
<i>Figura 130 Ventana para añadir información a la tarea</i> .....	191
<i>Figura 131 Mensaje informativo o de error</i> .....	191
<i>Figura 132 Mensaje con todos los errores producidos en la creación del curso</i> .....	192
<i>Figura 133 Mensaje para salir de la aplicación</i> .....	193
<i>Figura 134 Ventana para crear una nueva etiqueta de rol</i> .....	194
<i>Figura 135 Crear nueva tarea soporte</i> .....	195
<i>Figura 136 Introducción de opciones avanzadas</i> .....	196
<i>Figura 137 Editor gráfico. Secuenciación 1</i> .....	211
<i>Figura 138 Editor gráfico. Secuenciación 2</i> .....	212
<i>Figura 139 Editor gráfico. Secuenciación 3</i> .....	212
<i>Figura 140 Editor gráfico. Secuenciación 4</i> .....	213
<i>Figura 141 Editor gráfico. Secuenciación 5</i> .....	213
<i>Figura 142 Editor gráfico. Secuenciación 6</i> .....	214
<i>Figura 143 Editor gráfico. Secuenciación</i> .....	214
<i>Figura 144 Editor gráfico. Secuenciación 8</i> .....	215
<i>Figura 145 Editor gráfico. Secuenciación 9</i> .....	215
<i>Figura 146 Editor gráfico. Secuenciación 10</i> .....	216
<i>Figura 147 Editor gráfico. Secuenciación 1</i> .....	216
<i>Figura 148 Editor gráfico. Secuenciación 12</i> .....	217
<i>Figura 149 Editor gráfico. Secuenciación 13</i> .....	217
<i>Figura 150 Editor gráfico. Secuenciación 14</i> .....	218
<i>Figura 151 Editor gráfico. Secuenciación 15</i> .....	218
<i>Figura 152 Editor gráfico. Secuenciación 16</i> .....	219
<i>Figura 153 Editor gráfico. Secuenciación 17</i> .....	219
<i>Figura 154 Editor gráfico. Secuenciación 18</i> .....	220
<i>Figura 155 Editor gráfico. Secuenciación 19</i> .....	220
<i>Figura 156 Editor gráfico. Secuenciación 20</i> .....	221



# Índice de tablas

<i>Tabla 1. Condiciones del nivel B de IMS LD .....</i>	<i>40</i>
<i>Tabla 2 Expresiones del nivel B de IMS LD .....</i>	<i>41</i>
<i>Tabla 3 Acciones del nivel B de IMS LD .....</i>	<i>41</i>
<i>Tabla 4 Tabla comparativa de autoría .....</i>	<i>49</i>
<i>Tabla 5 Tabla comparativa de Reproductores.....</i>	<i>53</i>
<i>Tabla 6 Pruebas funcionales de la pestaña Info.General de CourseEditor v2.0 .....</i>	<i>124</i>
<i>Tabla 7 Pruebas funcionales de la pestaña Info.General de CourseEditor v2.0 .....</i>	<i>125</i>
<i>Tabla 8 Pruebas funcionales de la pestaña Info.General de CourseEditor v2.0 .....</i>	<i>125</i>
<i>Tabla 9 Pruebas funcionales de la pestaña Duración de CourseEditor v2.0.....</i>	<i>126</i>
<i>Tabla 10 Pruebas funcionales de la pestaña Objetivos de CourseEditor v2.0.....</i>	<i>128</i>
<i>Tabla 11 Pruebas funcionales de la pestaña U.D de CourseEditor v2.0 .....</i>	<i>128</i>
<i>Tabla 12 Pruebas funcionales de la pestaña T.Tareas de CourseEditor v2.0.....</i>	<i>129</i>
<i>Tabla 13 Pruebas funcionales de la pestaña Roles de CourseEditor v2.0 .....</i>	<i>130</i>
<i>Tabla 14 Pruebas funcionales realizadas en la pestaña Tareas de CourseEditor v2.0 .</i>	<i>131</i>
<i>Tabla 15 Pruebas funcionales de la pestaña UD-Tareas de CourseEditor v2.0.....</i>	<i>131</i>
<i>Tabla 16 Pruebas funcionales del editor gráfico de CourseEditor v2.0 .....</i>	<i>133</i>
<i>Tabla 17 Tabla de valor esperado de las pruebas estructurales en la pestaña Info.General de CourseEditor v2.0 .....</i>	<i>133</i>
<i>Tabla 18 Número de horas esperado. Pruebas estructurales en la pestaña Duración de CourseEditor v2.0 .....</i>	<i>134</i>
<i>Tabla 19 Fecha de inicio esperada. Pruebas estructurales en la pestaña Duración de CourseEditor v2.0 .....</i>	<i>135</i>
<i>Tabla 20 Fecha de fin esperada. Pruebas estructurales en la pestaña Duración de CourseEditor v2.0 .....</i>	<i>135</i>
<i>Tabla 21 Número de créditos esperado. Pruebas estructurales en la pestaña Duración de CourseEditor v2.0 .....</i>	<i>136</i>
<i>Tabla 22 Peso esperado. Pruebas estructurales en la pestaña Objetivos de CourseEditor v2.0 .....</i>	<i>137</i>

## ÍNDICE DE TABLAS

<i>Tabla 23 Valor de la tabla de U.D esperado. Pruebas estructurales en la pestaña U.D de CourseEditor v2.0 .....</i>	<i>138</i>
<i>Tabla 24 Tabla del valor de la tabla T.Tareas esperado. Pruebas estructurales en la pestaña T.Tareas de CourseEditor v2.0.....</i>	<i>139</i>
<i>Tabla 25 Valor de la tabla de roles esperado. Pruebas estructurales en la pestaña Roles de CourseEditor v2.0.....</i>	<i>140</i>
<i>Tabla 26 Valor de la tabla de tareas esperado. Pruebas estructurales en la pestaña Tareas de CourseEditor v2.0 .....</i>	<i>141</i>
<i>Tabla 27 Coste de los recursos hardware.....</i>	<i>152</i>
<i>Tabla 28 Desglose de costes en recursos humanos .....</i>	<i>153</i>
<i>Tabla 29 Coste total del proyecto .....</i>	<i>153</i>

# Capítulo 1

## Introducción

### 1.1 Introducción

La Tele-enseñanza o la Tele-educación se muestran como un mecanismo de educación a la hora de plantear una metodología determinada. Es el resultado de la aplicación de tecnología audiovisual y telemática en la enseñanza, algo que sin duda ha revolucionado el ámbito educativo.

Con el objeto de llevar a cabo un programa de formación basado en e-learning, se hace uso de plataformas o sistemas de software que permiten la comunicación e interacción entre profesores, alumnos y contenidos. Se tienen principalmente dos tipos de plataformas: las que se utilizan para impartir y dar seguimiento administrativo a los cursos en línea o LMS (*Learning Management Systems*) y, por otro lado, las que se utilizan para la gestión de los contenidos digitales o LCMS (*Learning Content Management Systems*).

Los sistemas actuales que ofrecen soluciones de e-learning utilizan diferentes contenidos digitales. Estos contenidos no pueden ser transferidos de forma fácil entre los distintos sistemas de e-learning. Así pues, surge la necesidad de definir formatos o normas denominados estándares de e-learning que sirvan de vehículo entre sistemas y que doten de flexibilidad a las soluciones existentes de e-learning.

Uno de los objetivos que se pretende conseguir mediante la definición de estándares de e-learning, es permitir intercambios de información entre diferentes LMS y creadores de cursos y que los distintos cursos y objetos de aprendizaje puedan ser reutilizados en otras herramientas y en otros LMS.

Entre las especificaciones de e-learning existentes, una de las más importantes es IMS-Learning Design (IMS-LD) cuyo objetivo es poder intercambiar recursos pedagógicos. Dicha especificación permite describir la metodología a aplicar en un determinado curso secuenciando tareas, sincronizando roles, definiendo diferentes caminos en función de posibilidades, etc.

Las herramientas de autor de IMS-LD existentes en la actualidad presentan dos problemas fundamentales. Por un lado, dichas herramientas son usualmente muy complejas de utilizar ya que abarcan todo el potencial de la especificación por lo que puede resultar complejo para usuarios sin experiencia previa. Por otro lado, las utilidades existentes para definir programaciones curriculares generalmente se basan en plantillas. Dichas plantillas son fijas y no permiten almacenar una metodología. Esto supone una pérdida de libertad y flexibilidad para el diseño de un curso.

Por tanto, dado que los profesores sin experiencia previa ni conocimientos en las especificaciones de e-learning suelen tener dificultades para adaptarse a este nuevo tipo de aprendizaje, y que las herramientas existentes que facilitan al docente la creación de cursos presentan los inconvenientes mencionados anteriormente, surgió la idea de la creación de CourseEditor, un programa software que permite a los usuarios la posibilidad de crear cursos de forma fácil y sencilla sin necesidad de que tengan ningún tipo de conocimiento de las especificaciones de e-learning existentes.

Este proyecto fin de carrera surge de la necesidad de mejorar la herramienta CourseEditor para que sea más usable y por tanto, los usuarios de la aplicación consideren la interacción más fácil e intuitiva. Además, se hace necesario añadir nueva funcionalidad que se adapte a la especificación IMS-LD y las nuevas necesidades de los usuarios para poder crear cursos más completos.

## 1.2 Objetivos

El objetivo fundamental de este proyecto de fin de carrera es, tal y como se ha mencionado anteriormente, la mejora de la herramienta CourseEditor, tanto en usabilidad como con la introducción de nuevas funcionalidades necesarias para hacerla más completa, todo ello, manteniendo la premisa de que la herramienta debe ser lo suficientemente sencilla de utilizar y a su vez permita la descripción de algunos de los aspectos más relevantes de la programación curricular de un curso completo.

Más detalladamente, los objetivos que se pretenden conseguir son:

- O1. Mejorar la interfaz de la herramienta de forma que sea más intuitivo su uso. Estos cambios incluirán:

- O1.1 La introducción de un menú en la parte superior izquierda, similar al de otras herramientas ofimáticas usadas habitualmente por profesores. Este menú deberá tener distintas funcionalidades, entre las que encuentran la posibilidad de ‘guardar’ un curso creado, ‘abrir’ un curso ya existente y mostrar un manual de ayuda.
- O1.2 Creación de un manual de ayuda, con un sistema de navegación que permita al usuario desplazarse por toda la ayuda en cualquier momento.
- O1.3 Revisión de los campos de introducción de texto y un ajuste del tamaño de la aplicación para poder visualizar mejor los contenidos introducidos en cada una de las tablas.
- O1.4 La posibilidad de definir un número libre de horas y créditos en los campos destinados a ello en lugar de seleccionarlos de un conjunto predefinido de valores tal y como ocurría en la versión anterior de la herramienta.
- O1.5 Eliminar las restricciones respecto a las fechas de inicio y finalización permitiendo su introducción mediante un calendario.
- O1.6 Permitir que los usuarios decidan la ruta en la que quieren guardar el curso creado, en lugar de guardarlo automáticamente en el mismo directorio en el que se encuentra el ejecutable de la herramienta.
- O1.7 Guardado de los ficheros generados por la herramienta en una carpeta comprimida en .zip, de forma que sea compatible con herramientas de simulación de cursos existentes en el mercado.
- O2. Corrección de errores de la herramienta entre los que se incluyen:
  - O2.1. Mejorar la definición de roles habilitando una nueva pestaña que permita añadir información más específica sobre estos.
  - O2.2. Errores existentes al añadir nuevas tareas, roles y secuenciaciones.
  - O2.3. En la versión anterior de la herramienta era necesario que el curso tuviera definida una tarea de soporte como requisito para permitir que el programa generara los ficheros XML compatibles con la especificación IMS-LD. En la nueva versión del programa se permite que se generen dichos ficheros XML sin tener en cuenta el precedente citado anteriormente.
- O3. Inclusión de nuevas funcionalidades, entre las que se incluyen:
  - O3.1. Inclusión de nuevas funcionalidades de nivel B recogidas en la especificación IMS-LD. Para ello se incluirán condiciones en la secuenciación que deben de ser lo suficientemente sencillas de definir para que un profesor que no tenga conocimientos técnicos de la especificación ni de e-learning pueda introducirlas sin dificultad.
  - O3.2. Mejora del editor gráfico para permitir guardar las secuencias generadas.

- O3.3. Realización de un tutorial para desarrolladores explicando cómo hacer modificaciones sencillas en la herramienta.
- O.4. Evaluación de la herramienta mediante la realización de pruebas estructurales y funcionales.

## 1.3 Estructura de la memoria

Para facilitar la lectura de la memoria, se incluye a continuación un breve resumen de cada capítulo:

**Capítulo 1: Introducción y objetivos.** En este capítulo se establece el propósito y los objetivos de todo el contenido del proyecto de fin de carrera.

**Capítulo 2: Estado del Arte.** En este capítulo se presenta la base teórica y las tecnologías relacionadas con el presente proyecto de fin de carrera. En primer lugar, se realizará una introducción al concepto de e-learning explicando sus principales características. Después, se presentarán las especificaciones más importantes existentes basadas en e-learning prestando especial atención a la especificación IMS-Learning Design dado que dicha especificación es la relacionada con este proyecto. Por último, se realizará un análisis de las herramientas existentes basadas en dicha especificación.

**Capítulo 3: Desarrollo del Proyecto.** En este capítulo se presenta una descripción detallada de todas las funcionalidades añadidas y los cambios realizados a la herramienta CourseEditor así como de los errores solucionados de la versión inicial del programa.

**Capítulo 4. Implementación del Proyecto.** En este capítulo se explica detalladamente las acciones llevadas a cabo para la realización de los cambios realizados en las clases Java existentes de la herramienta, así como de las nuevas clases añadidas.

**Capítulo 5: Evaluación de la herramienta.** En este capítulo se presentan los resultados de las pruebas realizadas sobre la herramienta CourseEditor. Se distinguen dos tipos de pruebas realizadas sobre la herramienta, pruebas funcionales y pruebas estructurales.

**Capítulo 6: Gestión del proyecto.** En este capítulo se presenta la metodología llevada a cabo para la realización de este proyecto de fin de carrera. Se detallan las fases de desarrollo en las que se ha llevado a cabo, así como la planificación temporal y el presupuesto del mismo.

**Capítulo 7: Conclusiones y trabajo futuro.** Se exponen las principales ideas, cuestiones y conclusiones derivadas de la realización del proyecto, así como las posibles líneas de investigación que se podrían llevar a cabo partiendo de este proyecto.

**Anexo 1: Tutorial.** Tutorial detallado sobre cómo realizar cambios sencillos en la herramienta.

**Anexo 2: Curso de prueba.** Descripción del curso que se ha llevado a cabo para probar la funcionalidad de la herramienta.

**Anexo 3: Secuenciaciones probadas.** Secuenciaciones del interfaz gráfico de la herramienta que han sido probadas.

**Glosario.** En este apartado se recogen los principales términos y conceptos técnicos utilizados en la memoria, con el objetivo de facilitar su comprensión al lector.

**Referencias.** En este apartado se reflejan las citas bibliográficas que se han consultado para la realización tanto del proyecto como de la memoria.





# Capítulo 2

## Estado del Arte

La herramienta CourseEditor, previamente desarrollada en otro proyecto de fin de carrera, y objeto de mejora e introducción de nuevas funcionalidades en el presente proyecto de fin de carrera, se desarrolló basándose en distintos modelos de aprendizaje. A su vez, la herramienta tiene como objetivo permitir a un docente que pueda diseñar un curso propio en función de sus necesidades. En primer lugar, se hará una introducción al diseño curricular. A continuación, se presentarán las especificaciones más relevantes basadas en e-learning. Por último, se hará una descripción detallada de IMS-LD, la especificación en la que se basa CourseEditor y se presentarán otras herramientas existentes que hacen uso de ella.

### 2.1 Diseño curricular

El diseño de un curso es el proceso mediante el cual un profesor elabora lo que se pretende alcanzar una vez impartido curso, así como la propuesta de acción para lograrlo.

Para ello, el docente debe plantearse una serie de intenciones educativas, es decir, qué va a enseñar, por qué, cómo y cuando. Por tanto, se hace necesario definir un programa del curso que debe incluir todos los elementos necesarios para que el alumno obtenga una visión global de lo que se le propone aprender, así como de la manera en que lo hará.

El diseño curricular puede entenderse como la metodología, las acciones y el resultado del diagnóstico, modelación, estructuración, y organización de los proyectos curriculares. [1]

Las distintas fases de las que se compone el diseño curricular son:

- **Análisis previo y diagnóstico de problemas y necesidades.**  
Esta fase consiste en un estudio del marco teórico de los principales aspectos que intervienen en el currículum. En esta fase se estudian los sujetos del proceso de enseñanza-aprendizaje, sus cualidades, motivaciones, conocimientos previos, preparación profesional, necesidades, intereses, etc. Los elementos que se obtienen de la exploración permiten caracterizar y evaluar la situación sobre la cual se debe diseñar el curso.
- **Modelación del currículum**  
En esta fase se lleva a cabo la tarea de determinación de los contenidos necesarios para alcanzar los objetivos y la descripción de la metodología a utilizar para el desarrollo curricular. El resultado de esta tarea debe quedar documentado definiendo los objetivos a alcanzar, así como la relación de los conocimientos, habilidades y las orientaciones metodológicas para la puesta en práctica.
- **Estructuración curricular**  
Esta tarea consiste en la secuenciación y estructuración de los componentes que intervienen en el proyecto curricular que se diseña. En este momento se determina el orden en que se va a desarrollar el proyecto curricular, el lugar de cada componente, el tiempo necesario y las relaciones de precedencia entre otros.
- **Organización para la puesta en práctica**  
Consiste en la previsión de todas las medidas necesarias para garantizar la puesta en práctica del diseño.
- **Evaluación curricular**  
Consiste en diseñar una evaluación a partir de los objetivos finales con el objeto de estudiar la efectividad de la puesta en práctica del proyecto curricular.

### 2.1.1 Aportaciones del diseño curricular al proyecto

Las distintas fases que se han visto en el apartado anterior proporcionan unas aportaciones muy valiosas a la herramienta CourseEditor. Una de las aportaciones más importantes es la de secuenciación de los elementos que conforman el curso.

Por otra parte, tal y como se ha mencionado en la fase de análisis previo se estudian las diferentes características de las personas que participan en el proceso de enseñanza-aprendizaje. Dentro de estas características estudiadas cabe las motivaciones, intereses y conocimientos previos, ya que dependiendo de estas características la secuenciación de los elementos podrían seguir diversos caminos. Por ejemplo, se podría plantear un examen inicial de forma que si un alumno tiene unos conocimientos previos superiores

que otro podría seguir un camino diferente en la secuenciación de elementos en función de la calificación que obtuvo en la prueba inicial. Por ello se ha decidido introducir una nueva funcionalidad en la herramienta, las condiciones, que permitirán la diversificación de caminos de aprendizaje en base a resultados previos u objetivos conseguidos obtenidos. La introducción de esta funcionalidad hace que la herramienta permita a los usuarios diseñar cursos más flexibles y completos que se adapten a las necesidades y características de los alumnos.

Finalmente, la modelación del curriculum se correspondería con la creación de un curso en sí, de forma que en la herramienta se permiten definir los objetivos y contenidos necesarios para llevarlo a cabo.

## 2.2 E-Learning

### 2.1.1 Introducción

El sector educativo ha encontrado en la tecnología un excelente medio para romper con las limitantes geográficas y temporales que los esquemas tradicionales de enseñanza-aprendizaje conllevan, revolucionando, y cambiando a la vez, el concepto de educación a distancia.

Existen diferentes definiciones del término e-learning. En su definición más amplia, tomando como referencia la raíz de la palabra traducida como “aprendizaje electrónico”, el término e-learning hace referencia a cualquier actividad educativa que utilice medios electrónicos para realizar todo o parte del proceso formativo.

La Dirección General de Telecomunicaciones definió e-learning en 1996 como: “El desarrollo del proceso de formación a distancia (reglada o no reglada) basado en el uso de las tecnología de la información y las telecomunicaciones, que posibilitan el aprendizaje interactivo, flexible y accesible a cualquier receptor potencial” [2]

En todas las definiciones que se pueden encontrar en la bibliografía relacionada con el e-learning, se hace mención explícita o implícita a la **tecnología** (plataformas, campus virtuales...), los **contenidos** (calidad y estructuración de los mismos se toman como elementos capitales para el éxito de una iniciativa de e-formación) y los **servicios** (siendo el elemento más variopinto que engloba la acción de los profesores, elementos de gestión, elementos de comunicación, elementos de evaluación, etc.). En función del grado de combinación de los elementos mencionados anteriormente se pueden obtener diferentes modelos de e-learning. [3]

El principio fundamental sobre el que se asienta el e-learning es la interacción. Este principio favorece la creación de procesos educativos más individualizados y colaborativos. Se pretende que el alumno tenga una mayor responsabilidad en su educación contribuyendo de forma activa en el proceso de enseñanza – aprendizaje.

El e-learning es, hoy en día, un complemento o en ocasiones un sustituto del aprendizaje presencial por lo que se hace necesario disponer de herramientas que permitan diseñar cursos de forma fácil e intuitiva y que a su vez permitan la reutilización de contenidos. A continuación se detallan algunas de las características del e-learning que han propiciado este hecho.

- El e-learning colabora en la desaparición de barreras espacio-temporales. Los métodos de aprendizaje presencial requerían acuerdo explícito respecto al momento en el que el aprendizaje iba a tener lugar así como un espacio físico para llevarlo a cabo. La introducción de métodos de e-learning en la enseñanza supone que los estudiantes puedan realizar cursos o formación sin la limitación de espacio y tiempo ya que los contenidos están disponibles en cualquier momento. De esta forma se optimiza al máximo el tiempo dedicado a la formación.
- Con el objetivo de proporcionar al alumnado una formación más flexible es necesario que se usen diversos métodos y recursos en el diseño de cursos que permitan al alumno desarrollar sus capacidades siguiendo diferentes itinerarios formativos en función de sus características y necesidades. Además, es importante que los participantes de un curso puedan desempeñar distintos papeles o roles en función de la tarea o actividad a desarrollar.
- En cuanto a los contenidos, el uso de herramientas basadas en e-learning permiten que la información y los recursos relacionados con el tema de estudio puedan ser actualizados de forma rápida, por lo que las herramientas de diseño de cursos deben de ser capaces de permitir al usuario actualizar o ampliar la información de forma sencilla y rápida.
- A diario surgen nuevas herramientas de soporte a la docencia que se incorporan a las plataformas e-learning como por ejemplo foros, chat, correo electrónico, etc y permiten que exista comunicación constante entre el alumno y profesor, o entre los participantes con diferentes tipos de roles en un mismo curso o programa, por lo que es necesario que las herramientas de diseño de cursos sean flexibles y proporcionen tareas de soporte de instalación y configuración de estas herramientas.

### 2.1.2 Elementos de E-Learning

En una plataforma o herramienta de e-learning se distinguen tres elementos principales que se describen a continuación. [4]

#### 2.1.2.1 Contenidos

Hace referencia al material didáctico empleado para la impartición de un curso. Los contenidos para e-learning se presentan en diversos formatos, en función de su adecuación a la materia tratada. Normalmente, estos contenidos son elementos multimedia e interactivos que permiten que el usuario avance por él mismo evaluando lo que aprende. Este método se da en cursos on-line o también denominados WBT (Web Based Training).

También es posible que los contenidos se ofrezcan al alumno mediante explicaciones en tiempo real llevadas a cabo en una sesión de aula virtual, es decir, una clase transmitida por videoconferencia en la que los alumnos pueden interactuar con el docente.

Cuando el material no puede ser ofrecido como material multimedia, se puede hacer llegar el material a los alumnos mediante la descarga de ficheros o documentos con el fin de que sean estudiados para luego realizar una puesta en común mediante la realización de actividades on-line como por ejemplo, la participación en foros o chats.

### **2.1.2.2 LMS**

Un Sistema de Gestión de Aprendizaje-LMS (*Learning Management System*) es un software que automatiza la administración de acciones de formación. Sus principales funciones son las siguientes:

- Gestionar los usuarios que toman parte de los cursos. Esta tarea se lleva a cabo mediante la inscripción de alumnos, control de aprendizaje de cada alumno, generación de informes, etc.
- Gestionar y lanzar los cursos. Esta tarea se lleva a cabo mediante la realización de un registro de la actividad del alumno, es decir, mediante el almacenamiento de los resultados de las pruebas que el alumno realiza o mediante el almacenamiento del tiempo que dedica el alumno a cada actividad y el tiempo que accede a cada material formativo disponible.
- Gestionar los servicios de comunicación. Esta tarea se lleva a cabo mediante la programación y gestión de foros de discusión, charlas, videoconferencias etc, cuando sean necesarios.

La mayoría de LMS están basadas en la Web para facilitar en cualquier momento y en cualquier lugar y a cualquier ritmo el acceso a los contenidos de aprendizaje y administración.

Habitualmente, el LMS lo emplean las organizaciones que poseen un volumen considerable de cursos diferentes y para ello se les hace necesario el uso de una herramienta tecnológica que les permita gestionarlos con efectividad y de forma práctica. Cada LMS ha creado sus propios formatos para estructurar contenidos y almacenar información, lo que trae como consecuencia la imposibilidad de interoperar entre las distintas plataformas, o crear productos únicos que funcionen con distintas plataformas.

### **2.1.2.3 Sistemas de comunicación.**

Los sistemas de comunicación pueden ser síncronos y asíncronos.

Un sistema de comunicación síncrono es aquel que ofrece comunicación en tiempo real entre los alumnos y el profesorado. Algunos ejemplos de sistemas de comunicación síncronos son chats, videoconferencias, pizarra virtual, etc.

Un sistema de comunicación asíncrono no permite la comunicación en tiempo real. Sin embargo, tiene la ventaja de que las aportaciones quedan grabadas para su posterior consulta. Algunos ejemplos de sistemas de comunicación asíncronos son los foros de discusión o correo electrónico.

### 2.1.3 Ventajas e Inconvenientes del E-Learning

A continuación, se expondrán algunas de las ventajas e inconvenientes que presentan los sistemas de e-learning. [5]

Entre las principales ventajas, se pueden encontrar:

- El alumno selecciona los temas que desea o precisa aprender en función de sus necesidades, por lo que la información brindada es totalmente personalizada, actualizada y de fácil distribución. Por tanto, las herramientas basadas en e-learning están focalizadas completamente en el alumno.
- El e-learning contribuye a la profesionalización desde el punto de vista de que permite al estudiante manejar sus tiempos de estudio. Esto supone que el estudiante tiene un mayor grado de responsabilidad cuando toma un curso de forma no presencial.
- Respeta los tiempos y los espacios de los participantes ya que éstos deciden el momento del día y los días de la semana que tomarán las clases.
- Supera los límites del aula, permite que el alumno pueda acceder a los contenidos las 24 horas del día, los 7 días de la semana desde cualquier dispositivo electrónico.
- Evita desplazamientos de los usuarios dado que los participantes ya no deberán movilizarse hasta el lugar en el que tiene lugar el curso por lo tanto, disminuyen los costes para el estudiante.
- Los contenidos necesarios para el correcto desarrollo de un curso se actualizan con mayor facilidad ya que gran parte de los materiales didácticos (textos, apuntes, papers etc.) están en formato digital y por este motivo el procedimiento para agregar, modificar o actualizar archivos es mucho más simple que si se manejan documentos impresos.
- El método de seguimiento electrónico es más eficaz y rápido porque se pueden implementar sistemas de alerta que avisen al instructor y a los alumnos de hitos o fechas importantes relativas al curso como por ejemplo, las fechas de entrega de trabajos, reuniones, exámenes, etc. También, es posible el uso de aplicaciones que permitan medir el tiempo que el alumno dedica a cada módulo así como aplicaciones que permitan al instructor llevar un control de los resultados de los ejercicios propuestos y calificaciones entre otros.
- Los sistemas basados en e-learning fomentan la comunicación entre el alumno y el profesor ya que se puede hacer uso de herramientas como el correo electrónico,

chats, foros, etc, de forma que exista un vínculo más rápido de comunicación entre ambos.

En cuanto a los inconvenientes encontrados en los sistemas basados en e-learning, se destacan los siguientes: [6]

- La planificación y desarrollo de un curso mediante teleformación requiere más inversión de trabajo que un curso presencial. Se necesita un equipo técnico de producción y de gestión.
- Se requiere más esfuerzo por parte del profesorado ya que aparte de ser expertos en su materia, deben ser expertos en el uso de las Tecnologías de la información y la comunicación (TIC) y en el aprovechamiento didáctico de las mismas.
- Para el alumnado, también se requiere un esfuerzo dado que es necesario que los estudiantes sean muy metódicos y organizados y tengan habilidad para el aprendizaje autónomo. Del mismo modo que el profesorado debe de tener un buen dominio de las tecnologías de la información y comunicación, es necesario que el alumnado también disponga de suficientes conocimientos de forma que no se vea retrasado su aprendizaje.
- En numerosos casos se hace necesario disponer de un dispositivo electrónico y de conexión a Internet.

### **2.1.4 Aportaciones de E-Learning al proyecto**

Tal y como se ha visto en los apartados anteriores, uno de los pilares del e-learning es la eliminación de barreras espacio – temporal, ya que el alumno decide en que momento lleva a cabo las actividades y tareas correspondientes al curso que realiza. Se pretende que CourseEditor sea una herramienta que permita a un docente describir la programación curricular de un curso, y en el caso de que se trate de un curso online, se permita flexibilidad en cuanto a los tiempos y los espacios de los participantes.

Otra de las bases en las que se asienta el e-learning es la posibilidad de definir itinerarios que se adapten a las características de los estudiantes. Los estudiantes pueden decidir el itinerario que deseen y pueden desempeñar un papel o rol en el desarrollo del curso. Esto se ha tenido muy en cuenta en la nueva versión de CourseEditor y por ello, se ha mejorado la definición de roles y la secuenciación de tareas de forma que cada usuario pueda desempeñar un rol en un momento determinado y pueda seguir un itinerario acorde a la definición del curso.

De los elementos de e-learning vistos, cabe destacar por su relación al proyecto los sistemas de gestión de aprendizaje (LMS). La importancia de los LMS reside en que estos pueden cargar contenidos y metodologías en diferentes formatos digitales. Se hace necesario definir un conjunto de reglas que permita que los contenidos puedan ser utilizados en diferentes plataformas. Este conjunto de normas se denomina ‘especificación de e-learning’ y se estudiará en detalle en el siguiente apartado. CourseEditor almacena la información siguiendo la especificación de e-learning IMS-LD.

De este modo, la información que se introduzca en la herramienta y los cursos diseñados con la misma podrán ser reutilizados en otros LMS.

Por último, uno de los inconvenientes del e-learning es el mayor esfuerzo que se requiere por parte del profesorado, ya que se espera que conozcan las especificaciones de e-learning y sean capaces de diseñar cursos de acuerdo a ellas. En la nueva versión de CourseEditor se ha mejorado la interfaz y el proceso de creación de cursos de forma que sea más fácil e intuitiva para el personal docente que no es experto en la especificación.

## 2.3 Especificaciones de E-Learning

### 2.2.1 Introducción

Un estándar es un patrón o una norma documentado en detalle y ratificado por una autoridad respetada de su campo sobre cómo realizar algo. Algunas de las autoridades más relevantes encargadas de publicar estándares son ISO (*International Standards Organisation*) [7], BSI (*British Standards Institute*) [8], CEN (*Centre Europeande Normalisation*) [9] o IEEE (*Institute of Electrical and Electronics Engineers*) [10].

Una especificación es el paso previo al estándar, creado por alguna compañía u organismo, que no ha sido ratificado todavía por ninguna autoridad, y que suele utilizarse de manera provisional pero suficientemente respaldada. Así pues, una especificación puede considerarse como un conjunto de declaraciones detalladas y exactas de los requisitos funcionales y particularidades de algo que quiere construirse, instalarse o manufacturarse.

En referencia al e-learning, las especificaciones y los estándares educativos tratan de regular cómo se realizan los procesos de enseñanza de modo que la información del sistema y los contenidos puedan ser reutilizables, interoperables e intercambiables entre distintas plataformas de enseñanza. De esta forma, una de las principales funciones de los estándares es facilitar la durabilidad y la reutilización en el tiempo de las aplicaciones y de la interoperabilidad, es decir, facilitar el intercambio de los contenidos entre diversas plataformas y sistemas. [11]

Distintas organizaciones y empresas relacionadas con el mundo del software, la capacitación y la educación están trabajando en la creación de estándares y especificaciones que logren crear plataformas, materiales y recursos interoperables entre distintos LMS, además de la creación de nuevos productos que se dediquen a generar estos recursos interoperables, o que ayuden a transformar los existentes. Con esto se consigue que la inversión tanto en tiempo como en costes económicos invertida en la creación de un curso no se pierde si se cambia de LMS, así como fomentar el intercambio y reutilización de contenidos.



### 2.2.2 Institute for Electrical and Electronic Engineers Learning Technology Standards Committee (IEEE LTSC)

El IEEE es un conocido cuerpo multinacional que desarrolla estándares internacionales para sistemas eléctricos, electrónicos, computacionales y comunicacionales. IEEE está organizado en distintos comités que se reúnen y analizan las distintas tecnologías. A partir de estos análisis generan como resultado una especificación o recomendación en forma de estándar. Uno de éstos es el Learning Technology Standards Committee (LTSC) o Comité para los Estándares de la Tecnología del Aprendizaje. [12]

Mediante sus actividades de publicación técnica, conferencias y estándares basados en consenso, el IEEE produce más del 30% de la literatura publicada en el mundo sobre ingeniería eléctrica, computación y telecomunicaciones.

### 2.2.3 Advanced Distributed Learning (ADL)

ADL nace como una organización auspiciada por el gobierno de Estados Unidos (el consumidor más grande de e-learning en el planeta), con el objetivo de investigar y desarrollar especificaciones que fomenten la adopción y el avance del e-learning.

Su principal objetivo es buscar mecanismos para asegurar educación y materiales de capacitación de alta calidad que puedan ajustarse las necesidades de cada institución. Como consecuencia de estas investigaciones y recomendaciones, ADL se está transformando en un gran impulsor para convertir las especificaciones en estándares.

La publicación más importante de ADL es el Modelo de Referencia de Objetos de Contenido Compartibles (Shareable Content Object Reference Mode), también conocido como SCORM. [13]

La definición del modelo SCORM, así como su evolución y las distintas decisiones de diseño tomadas durante el proceso de especificación, se basan en 6 principios esenciales:

- Accesibilidad ó posibilidad de localizar y acceder a componentes instruccionales desde una ubicación remota.
- Adaptabilidad ó posibilidad de adaptar la enseñanza a distintas necesidades individuales u organizacionales.
- Asequibilidad ó posibilidad de aumentar la eficiencia y la productividad reduciendo el tiempo y el coste invertidos en la enseñanza.
- Durabilidad ó posibilidad de resistir la evolución de la tecnología y futuros cambios sin que sea necesario incurrir en rediseños, reconfiguraciones o recodificaciones excesivamente costosas.

- Interoperabilidad ó posibilidad de tomar componentes instruccionales desarrollados en una ubicación determinada y poder hacer uso de ellos en otra ubicación y otro conjunto de herramientas y plataformas.
- Reusabilidad ó flexibilidad para incorporar componentes instruccionales en múltiples contextos y aplicaciones.

La especificación de SCORM se centra en 3 aspectos. Estos aspectos se han publicado en un documento técnico distinto. Los tres documentos que especifican los detalles relativos a cada aspecto de la especificación son los siguientes:

- Modelo de Agregación de Contenido (*Content Aggregation Model, CAM*).  
En este manual se describen los distintos tipos de objetos de contenido permitidos dentro de la especificación y detalla los mecanismos que se deben seguir para conseguir la interoperabilidad entre distintos LMS.
- Entorno de Tiempo de Ejecución (*Run-Time Environment, RTE*)  
En este manual se estandariza el proceso de ejecución de los contenidos para garantizar la interoperabilidad entre distintos LMS. Por ello, este manual define el proceso de ejecución y los mecanismos de comunicación que tanto el LMS como el propio contenido deben emplear.
- Secuenciamiento y Navegación (*Sequencing and Navigation, SN*): Este manual define los mecanismos para que los LMS puedan concatenar las actividades educativas de modo consistente. El manual recoge los eventos que pueden ser generados por los alumnos o por el sistema y que deben ser procesados por el LMS para decidir cual es el recurso educativo que debe ser servido a continuación.

SCORM ha tenido y tiene un gran impacto en el campo del aprendizaje a través de Internet dado que tanto la industria como el mundo académico han reconocido el liderazgo de ADL como entidad de referencia a la hora de valorar la calidad de los procesos de aprendizaje. En la actualidad la compatibilidad con SCORM es el principal punto de encuentro entre todas las organizaciones implicadas en el campo del aprendizaje asistido por ordenador. [14]

### 2.2.4 Aviation Industry CBT Committee (AICC)

AICC es un grupo internacional de Profesionales del entrenamiento y capacitación basada en tecnología. AICC fue creado en 1988. AICC se crea cuando Internet y la web no existían y por tanto sólo se hablaba del CBT (*Computer-Based Training*). En 1992 decidieron crear un comité que desarrollase una normativa para sus proveedores de formación basada en computador. De este modo garantizaban la armonización de los requerimientos de los cursos, así como la homogeneización de los resultados obtenidos de los mismos.

AICC ha publicado varias guías, incluyendo algunas relacionadas con hardware y software. Sin embargo, las que han tenido una mayor repercusión son las dedicadas a la CMI (*Computer Managed Instruction*).

### 2.2.5 IMS Global Learning Consortium

IMS (Instruction Management Systems) es un consorcio que agrupa a vendedores, productores, implementadores y consumidores de e-learning, y que se enfoca completamente a desarrollar especificaciones en formato XML. Entre sus miembros se encuentra Microsoft, Apple, ORACLE, WebCT, Blackboard y Boeing. [15]

Tal y como se ha mencionado anteriormente, IMS (Global Learning Consortium, Inc) es el principal promotor y desarrollador de especificaciones abiertas orientadas a la enseñanza electrónica. Su objetivo es que a partir de estas especificaciones se consiga la interoperabilidad de aplicaciones y servicios en la enseñanza electrónica para que los autores de contenidos y de entornos puedan trabajar conjuntamente.

Las especificaciones IMS cubren un amplio rango de características que se persiguen hacer interoperables entre plataformas.

La especificación IMS establece XML (*eXtensible Markup Language*) como el lenguaje seleccionado para realizar los intercambios de información entre distintas plataformas. XML es un lenguaje de marcas desarrollado por el *World Wide Web Consortium* (W3C) [16]. Aparte de su aplicación para Internet, este lenguaje se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Es por esto por lo que la especificación IMS hace uso de él para hacer interoperables las diferentes plataformas.

En la Figura 1 se muestra la arquitectura establecida en la especificación IMS.



*Figura 1 Arquitectura de la especificación IMS*

IMS ha publicado numerosas especificaciones, cada una de ellas enfocada a una necesidad distinta del proceso de enseñanza. Todas las especificaciones de IMS se detallan en tres documentos: [17]

- Guía de Implementación y Consejos  
Es el documento más narrativo de los tres. En él se incluyen: la forma de uso de la especificación, la relación con otras especificaciones, y cualquier tipo de información

variada que pueda servir de ayuda. Suele ser el documento que se usa para iniciarse en la especificación.

- Modelo de Información  
Documento que muestra la estructura de datos completa. Normalmente posee una tabla detallada de cada uno de los elementos de la especificación. En ella se enumeran las propiedades de los elementos tales como el nombre, la multiplicidad, el tipo o si son obligatorios o no.
- Documento de Enlace  
Documento que ofrece la forma de representar la estructura de datos de la especificación en XML. Muestra el árbol XML con cada uno de sus elementos y sus atributos.

Algunas de las especificaciones desarrolladas por IMS y que se encuentran detalladas en los distintos documentos mencionados anteriormente son: [18]

- IMS Learning Resources Meta - data Specifications  
Establece una forma uniforme para describir los recursos de aprendizaje de manera que éstos puedan ser fácilmente encontrados, por medio del uso de herramientas de búsqueda que sean capaces de interpretar metadatos.
- MS Enterprise Specification  
Está orientada a aplicaciones y servicios administrativos que necesitan compartir datos sobre los alumnos, cursos, rendimiento a través de sistemas operativos, plataformas, interfaces de usuario.
- IMS Learner Profiles Specification  
Esta especificación establece formas de organizar la información del alumno de manera que los sistemas de aprendizaje puedan ser más responsivos a las necesidades específicas de cada usuario.
- IMS Reusable Competency Definition Specification  
Esta especificación define un modelo de información para describir, referenciar e intercambiar definiciones de competencias, principalmente en el contexto del aprendizaje.
- IMS Guidelines for Developing Accessible Learning Applications  
Se trata de una serie de guías desarrolladas por el Grupo de Trabajo sobre Accesibilidad de IMS para las comunidades de aprendizaje distribuido.
- IMS Digital Repositories  
Esta especificación busca integrar el aprendizaje online con los recursos de información a través de bodegas o depósitos digitales para almacenar colecciones digitales de documentos.
- IMS Learning Design Specification  
Esta especificación está orientada a describir el Diseño Instruccional y Diseño de Aprendizaje que acompañará a un curso online. La herramienta implementada en este

proyecto de fin de carrera sigue esta especificación. Por este motivo se desarrollará en detalle las características de dicha especificación en el apartado siguiente.

## 2.3 IMS-LD Learning Design

### 2.3.1 Introducción al estándar IMS-LD

Un Diseño de Aprendizaje (Learning Design) es definido en la especificación IMS como *“una descripción de un método que permite a los alumnos alcanzar ciertos objetivos de aprendizaje por medio del desarrollo de ciertas actividades de aprendizaje en un cierto orden en el contexto de un cierto ambiente de aprendizaje”*.

La especificación IMS Learning Design (IMS LD) es un lenguaje de modelado educativo cuyo objetivo es definir una estructura semántica para anotar los procesos de enseñanza y aprendizaje, y así convertirlos en entidades reutilizables entre diferentes cursos y aplicaciones. Para ello describe procesos que, con la intención de alcanzar un objetivo de aprendizaje, agrupan las actividades que realizarán alumnos y profesores, indicando en qué momento, con qué recursos didácticos y servicios, y bajo qué condiciones lo harán. [18]

IMS-LD define formalmente una estructura para describir cualquier diseño de enseñanza y aprendizaje. Concretamente, esta especificación pretende:

- Describir totalmente el proceso de enseñanza y aprendizaje de una Unidad de Aprendizaje (UoL).
- Flexibilidad pedagógica: Anotar el significado y la funcionalidad pedagógica de los elementos (e.g. actividades, objetivos, métodos, etc.) de una UoL.
- Personalización: Marcar aspectos de personalización para que las actividades puedan adaptarse a las preferencias, conocimientos previos o necesidades educativas de los usuarios.
- Formalización: Formalizar la descripción del diseño instruccional para poderlo procesar automáticamente.
- Reproductividad: Describir de manera abstracta el diseño instruccional para poderlo repetir en diferentes condiciones y con diferentes personas.
- Reusabilidad: Identificar, descontextualizar e intercambiar elementos de aprendizaje y reutilizarlos en otros contextos.
- Interoperabilidad: Fomentar el intercambio y uso de información entre diferentes aplicaciones compatibles con IMS LD.

Esta especificación permite crear planes de aprendizaje para distintos alumnos de forma flexible. Dichos planes pueden ser leídos por LMS denominados “player”. Así pues, un player de IMS-LD es una aplicación capaz de cargar un archivo en formato

IMS-LD. Entre las acciones que pueden realizar en el “player” destaca la asociación de roles y usuarios y la secuenciación de las diferentes actividades que están definidas en el curso en función del rol que desempeña cada usuario.

Según la especificación IMS-LD el proceso de aprendizaje se lleva a cabo de la siguiente manera. Un estudiante adquirirá un rol en el proceso de enseñanza – aprendizaje. En función del rol adquirido se desarrollaran distintas actividades para conseguir ciertos objetivos. Las actividades realizadas bajo esta especificación podrán ser actividades de aprendizaje o actividades de soporte. Este proceso se corresponde con el modelo diseñado en la herramienta CourseEditor, en la cual se espera que las diferentes tareas que se definan sean realizadas por usuarios que desempeñen un rol también definido.

La especificación IMS-LD cuenta con tres niveles de complejidad. La funcionalidad de la versión inicial de CourseEditor cubría las características descritas por el nivel A. La nueva versión de la herramienta amplía su funcionalidad añadiendo condiciones y expresiones definidas en el nivel B. Dada la importancia de los niveles de complejidad establecidos por la especificación IMS-LD en la aplicación desarrollada en el presente proyecto de fin de carrera se realizará una descripción de dichos niveles en los apartados siguientes.

### 2.3.2 Niveles de IMS LD

La especificación IMS LD se divide en tres niveles de complejidad creciente. Cada nivel se construye sobre el anterior añadiendo nuevas características que permiten añadir sofisticación a los diseños educativos. [18]

- Nivel A
- Nivel B
- Nivel C

#### 2.3.2.1 Nivel A

El nivel A de la especificación permite describir diseños educativos *no adaptativos*, en el sentido de que el método pedagógico siempre exhibirá el mismo comportamiento, independientemente del resultado de las distintas actividades. Este nivel incluye los principales elementos estructurales (recursos y servicios, roles, actividades, etc.) y dinámicos (aquellos correspondientes al método pedagógico) de IMS LD. El diseño de la especificación en este nivel consta de:

- **Título:** el cual describe diseño educativo.
- **Objetivos de aprendizaje** (opcionales): Estos objetivos harán referencia a los logros que se espera que consigan los alumnos que cursen la unidad.
- **Prerrequisitos** (opcionales): Se refieren a las condiciones que se aconseja a los alumnos cumplir para la realización del curso.

- **Componentes del diseño educativo:** Enumeran los participantes, las actividades y los materiales involucrados en el proceso de aprendizaje.
- **Método pedagógico:** Es la estructura que sigue la unidad de aprendizaje.
- **Metadatos:** Es la información adicional que permiten que el curso sea reutilizado por terceros. Dichos metadatos se añaden a los contenidos y describen distintas características semánticas de los mismos.

### 2.3.2.2 Nivel B

IMS LD nivel B añade un nuevo tipo de componente educativo a los diseños: las *propiedades*. Dichas propiedades almacenan información relevante que se produce durante la ejecución del método educativo (por ejemplo, puntuación del alumno en un test, si la tarea definida es entregable o no, y si a su vez, hay flexibilidad o no en la fecha de entrega de la misma).

Esto supone que el itinerario del proceso de aprendizaje puede variar en función de si se cumplen unas condiciones u otras favoreciendo la personalización e individualización de la enseñanza. A continuación se detallan las principales características de los mencionados componentes.

#### 2.3.2.2.1 Propiedades

Las propiedades que se añaden, se dividen en dos tipos: locales y globales.

- Las propiedades locales son aquéllas que se crean y se mantienen únicamente dentro de cada ejecución de la unidad de aprendizaje.

A su vez, dentro de estas propiedades se pueden distinguir tres subtipos: propiedades locales personales (cuyo valor puede ser distinto para cada participante), propiedades locales de rol (cuyo valor es el mismo para todos los miembros de un mismo rol, pero que puede variar entre distintos roles) y propiedades locales generales (cuyo valor es el mismo para todos los participantes)

Este tipo de propiedades son las que se han decidido introducir en la introducción de nueva funcionalidad del presente proyecto de fin de carrera, ya que se han considerado que son las más apropiadas para la creación de cursos más flexibles.

- Las propiedades globales son aquellas que se crean y se mantienen en el entorno en el que se ejecutan las unidades de aprendizaje. El valor de dichas propiedades se mantendrá, por tanto, a las distintas ejecuciones.

Dentro de estas propiedades se puede distinguir dos subtipos: propiedad global personal (cuyo valor varía para cada posible usuario) y propiedad global general (cuyo valor es el mismo para todos los usuarios).

### 2.3.2.2.2 Condiciones

Las condiciones son reglas que, en función del cumplimiento de una determinada afirmación, permite actualizar los valores de las propiedades. Las condiciones se evalúan o bien al comienzo de la ejecución de la unidad o bien cuando el valor de una propiedad haya cambiado. El elemento `<conditions>` será añadido al contenido del elemento `<method>` del modelo de nivel A. La descripción de cada elemento se detalla en la Tabla 1.

Expresión	Descripción
<code>&lt;if&gt;</code>	Se utiliza para evaluar una determinada expresión. Su resultado será true (verdadero) o false (falso).
<code>&lt;then&gt;</code>	Esta condición es ejecutada cuando el resultado de la expresión <code>&lt;if&gt;</code> es verdadera.
<code>&lt;else&gt;</code>	Esta condición es ejecutada cuando el resultado de la expresión <code>&lt;if&gt;</code> es falsa.

Tabla 1. Condiciones del nivel B de IMS LD

### 2.3.2.2.3 Expresiones

IMS LD nivel B incluye mecanismos para describir *expresiones* cuya evaluación da lugar a *valorse*. Gracias a estas expresiones es posible, por ejemplo, comprobar que actividad debe realizar un alumno en función de la nota obtenida en un test o en función de si una actividad anterior ha sido completada. Las expresiones utilizadas son las que se pueden observar en la Tabla 2.

Expresión	Descripción
<code>&lt;is-member-of-role&gt;</code>	Verdadera cuando el usuario pertenece al rol referido y falsa en otro caso.
<code>&lt;is&gt;</code>	Permite comprobar si dos valores son iguales.
<code>&lt;is-not&gt;</code>	Permite comprobar si dos valores son distintos.
<code>&lt;and&gt;</code>	Verdadera cuando lo son todos sus argumentos.
<code>&lt;or&gt;</code>	Verdadera cuando lo es alguno de sus argumentos.
<code>&lt;sum&gt;</code>	Suma todos sus argumentos.
<code>&lt;subtract&gt;</code>	Resta a su primer argumento el segundo.
<code>&lt;multiply&gt;</code>	Multiplica ambos argumentos.
<code>&lt;divide&gt;</code>	Divide el primer argumento entre el segundo.
<code>&lt;greater-than&gt;</code>	Permite comprobar si el valor del primer argumento es mayor que el valor del segundo.
<code>&lt;less-than&gt;</code>	Permite comprobar si el valor del primer argumento es menor que el valor del segundo.
<code>&lt;users-in-role&gt;</code>	Restringe la aplicabilidad de la expresión a todos aquellos usuarios que pertenecen al rol indicado.
<code>&lt;no-value&gt;</code>	Verdadera sobre propiedades que no tienen valor asignado.
<code>&lt;time-unit-of-learning-started&gt;</code>	Su valor es el momento en el que comenzó a ejecutarse la unidad.
<code>&lt;daytime-activity-&gt;</code>	Su valor es el momento en el que comenzó a ejecutarse la



<b>started&gt;</b>	unidad.
<b>&lt;current-daytime&gt;</b>	Su valor es el día y la hora actual.
<b>&lt;complete&gt;</b>	Permite comprobar si un determinado componente (actividad, actuación, acto o guión) ha finalizado.
<b>&lt;not&gt;</b>	Permite comprobar el incumplimiento de una determinada afirmación.

*Tabla 2 Expresiones del nivel B de IMS LD*

#### 2.3.2.2.4 Acciones

IMS LD incluye también mecanismos que permiten expresar acciones. Estos mecanismos se detallan en la Tabla 3 y principalmente se usan para determinar que actividades o componentes educativos deben de ser visibles a un alumno en función de las condiciones de visibilidad establecidas para dicho componente. También, es posible modificar el valor de una propiedad definida mediante el uso de acciones.

<b>Atributo</b>	<b>Descripción</b>
<b>show</b>	Permite mostrar algún componente educativo o recurso.
<b>hide</b>	Permite ocultar algún componente educativo o recurso.
<b>change-property-value</b>	Permite cambiar el valor de una propiedad.

*Tabla 3 Acciones del nivel B de IMS LD*

#### 2.3.2.3 Nivel C

El nivel C añade un mecanismo soportado por notificaciones que permite el envío de mensajes y la configuración de actividades dependiendo de la ocurrencia de determinados eventos. Se mejora así, la comunicación tanto síncrona como asíncrona entre el docente y el alumno.

Las notificaciones de nivel C incluyen:

- Una identificación del/los destinatarios a los que va dirigida la notificación.
- Opcionalmente, una referencia a una actividad de aprendizaje o de soporte.
- Opcionalmente, un campo de asunto con una breve descripción de la notificación.

#### 2.3.3 Principales limitaciones de la especificación

Las principales limitaciones de la especificación siguen siendo la dificultad de uso por parte de profesores que no tienen suficientes conocimientos de la especificación.

Muchas de las herramientas existentes que utilizan la especificación han intentado minimizar esta dificultad, pero normalmente siguen siendo demasiado complejas, porque normalmente constan de una serie de formularios en los que se solicita al usuario rellenar el valor de etiquetas que forman parte de la especificación de forma que un profesor sin amplios conocimientos de la la misma y especialmente si tampoco posee amplios conocimientos informáticos puede encontrar especialmente difícil este método.

CourseEditor fue desarrollada desde su primera versión de forma que la definición de cursos fuera hecha de la manera más simple posible de manera que cualquier usuario pudiera utilizarla para definir sus cursos de manera sencilla.

El desarrollo de la nueva versión de la herramienta llevada a cabo en el presente proyecto de fin de carrera ha seguido la premisa de mantener la sencillez de definición de cursos, y a su vez, mejorar la usabilidad y utilizadad de la misma.

La principal aportación al presente proyecto de fin de carrera la proporciona la especificación IMS-Learning Design, ya que la herramienta se basa en ella para la creación del fichero XML que define el curso creado. No obstante, el resto de especificaciones también tienen su aportación en cuanto a sus principales características se refiere, es decir, el curso creado se realiza de forma que pueda ser reutilizable, interoperable e intercambiable entre distintas plataformas de enseñanza.

## 2.4 Herramientas que utilizan la especificación IMS-LD

En este apartado se realiza una breve descripción de algunas de las herramientas existentes que utilizan la especificación IMS-LD. Estas herramientas se pueden clasificar en tres tipos:

- Herramientas de Autoría.
- Motores de Ejecución.
- Reproductores (o “players”).

### 2.4.1 Herramientas de autoría

Una herramienta de autoría da la capacidad a un cliente de generar objetos de aprendizaje de manera autónoma y después gestionarlos en un sistema de administración del conocimiento.

Las herramientas de autor o metamedios son aplicaciones que permiten un trabajo multimedia y constructivista para generar un entorno de aprendizaje dinámico. Dentro de las funcionalidades que este tipo de herramientas presentan se puede destacar la

posibilidad de crear actividades o pequeñas aplicaciones desde la misma herramienta. [19]. Las herramientas de autor permiten interconectar componentes que se puedan adecuar a los objetivos y habilidades que se pretendan desarrollar.

A continuación se describen las principales características de las herramientas más relevantes de este tipo:

- **CopperAuthor** (Figura 2) es una herramienta [20] que permite al usuario construir un diseño educativo y navegar sobre su estructura mediante una interfaz basada en tablas. Esta herramienta fue desarrollada en paralelo con el motor de ejecución *CopperCore* y sólo permite el desarrollo de diseños educativos IMS-LD de nivel A. La mayor desventaja de esta herramienta es que los nombres que aparecen en la interfaz que el usuario debe rellenar, son directamente los nombres de las etiquetas de la especificación IMS-LD, lo cual puede resultar extremadamente complicado para profesores que no conocen a fondo la especificación.

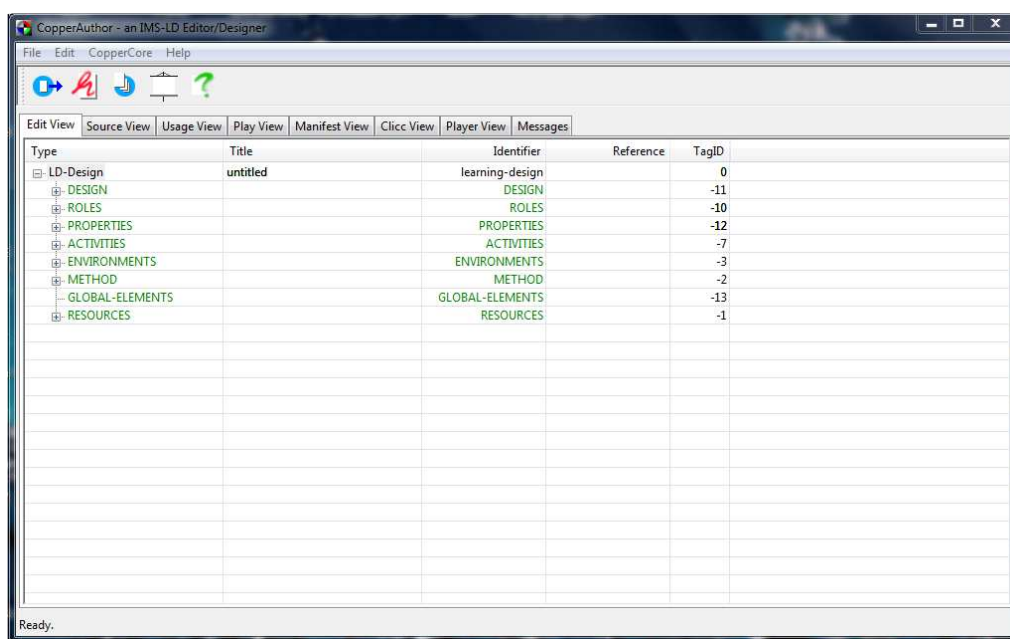


Figura 2 Interfaz de CopperAuthor

- **RELOAD** (*Reusable E-Learning Object Authoring & Delivery*) (Figura 3) es un editor desarrollado bajo un proyecto patrocinado por la iniciativa JISC del Reino Unido. Los principales objetivos de este proyecto son facilitar la creación, compartición y uso de los objetivos de aprendizaje y mejorar la gama de enfoques pedagógicos realizables. Para ello la herramienta usa planes de lecciones y presenta una interfaz que permite la edición de una UoL mediante la interacción con múltiples formularios y estructuras en forma de árbol por lo que resulta muy intuitiva y organizada al usuario. Se pueden crear diseños educativos IMS-LD de nivel A, B y C.

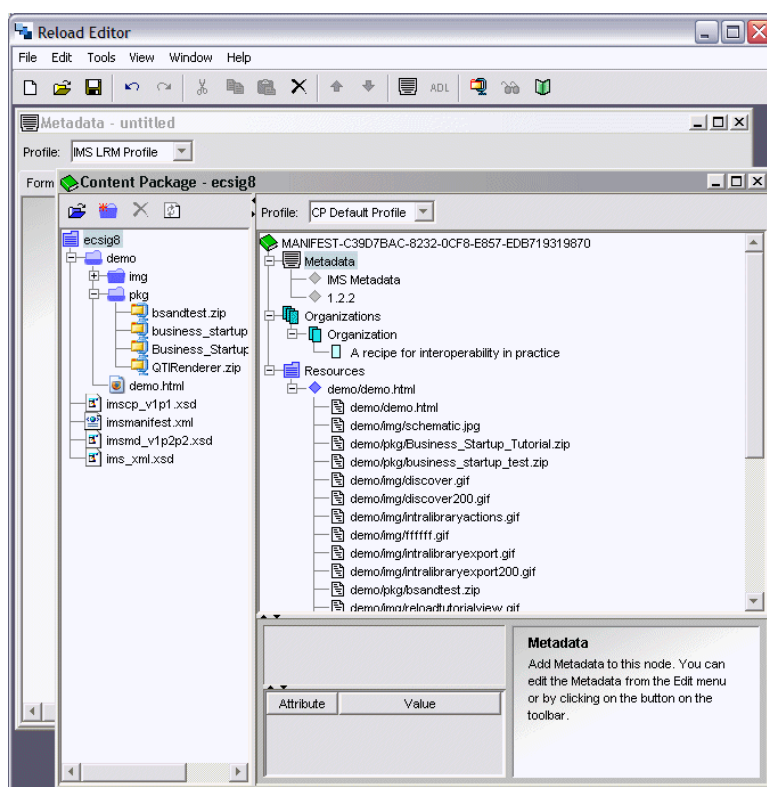


Figura 3 Interfaz de Reload Editor

- **CoSMoS** (*Collaboration Script Modelling System*) (Figura 4) es una herramienta de autoría ideada inicialmente para dar soporte a la formalización de procesos de aprendizaje colaborativos. Posteriormente la herramienta fue modificada para dar soporte a los conceptos de IMS-LD. Esta herramienta permite editar UoL mediante la navegación sobre la estructura de árbol en la que se representa. Además, permite la edición de los conceptos asociados a una UoL mediante el uso de formularios.

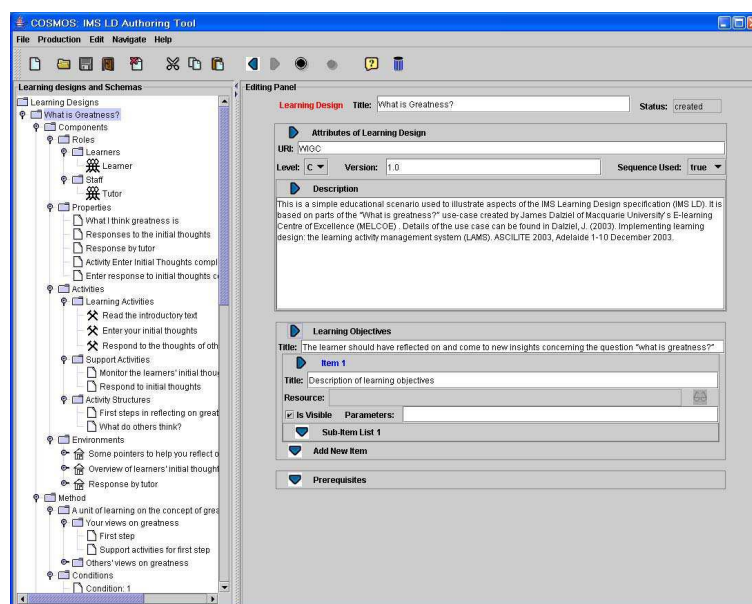


Figura 4 Interfaz de CoSMoS

- **MOT+** (Figura 5). Se trata de una herramienta desarrollada en el centro de investigación LICEF de Canadá, cuyo objetivo inicial era estructurar mapas conceptuales para la representación de conocimiento en diversos dominios. Posteriormente, MOT+ fue extendida para soportar la edición de IMS-LD, de manera que la herramienta permite representar y editar los conceptos que están definidos en dicha especificación. Con esta herramienta se pueden crear diseños educativos IMS-LD de nivel A y se están realizando trabajos para soportar los niveles B y C.

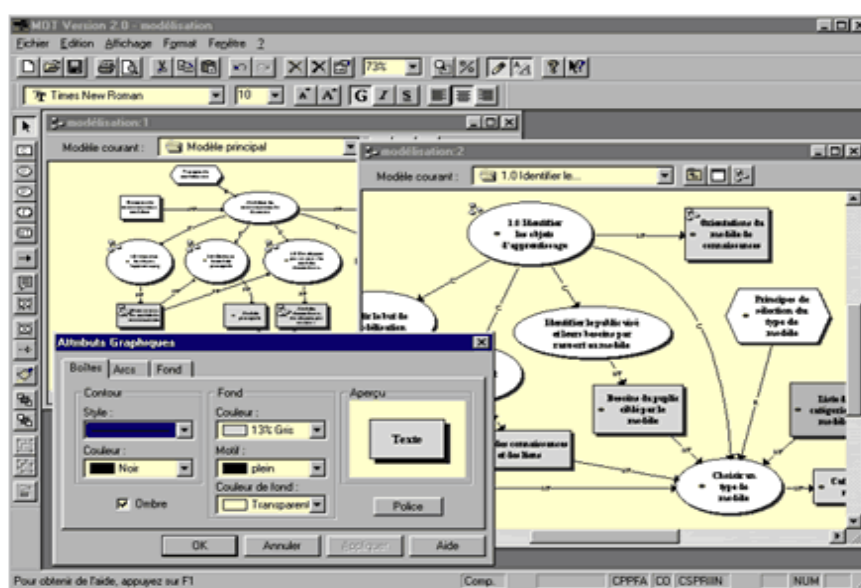


Figura 5 Interfaz de MOT+

- **ASK-LDT** (*Advanced e-Services for the Knowledge Society Research Unit*) (Figura 6) es una herramienta que se basa en la interconexión de tareas de aprendizaje siguiendo la especificación IMS-LD. Esta herramienta dispone de diferentes tipos de actividades predefinidas como por ejemplo, lecciones, discusiones o exámenes. Cada tipo de actividad dispone de una notación gráfica mediante la que es posible definir el flujo entre ellas. Con esta herramienta se pueden crear diseños educativos IMS-LD de nivel A y parcialmente de nivel B.

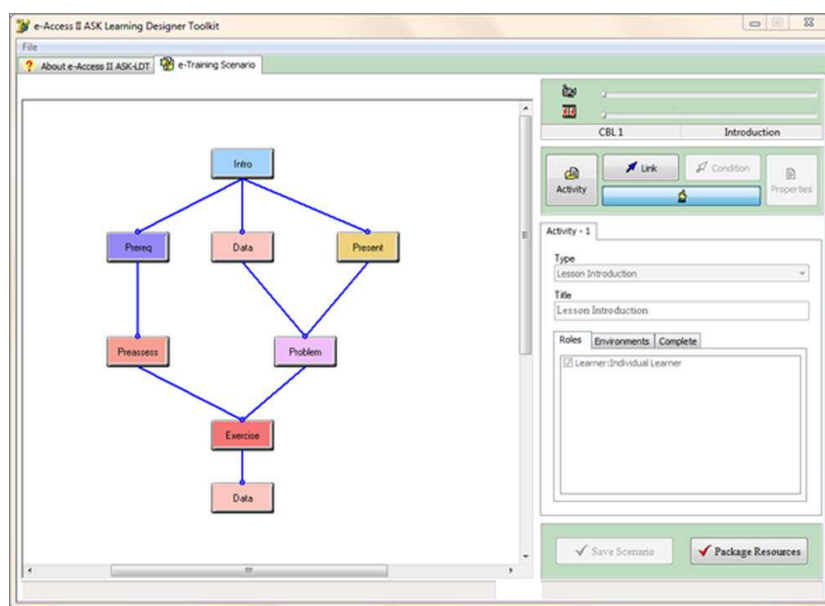


Figura 6 Interfaz de ASK-LTD

- **Learning Activity Management system (LAMS)** (Figura 7) es una herramienta que permite el diseño, gestión y distribución de actividades colaborativas para e-learning. Esta herramienta proporciona un entorno de autoría visual muy intuitiva que permite crear secuencias de actividades de aprendizaje. Las actividades incluyen una amplia gama de tareas, tanto individuales como en pequeños grupos ó clases enteras.

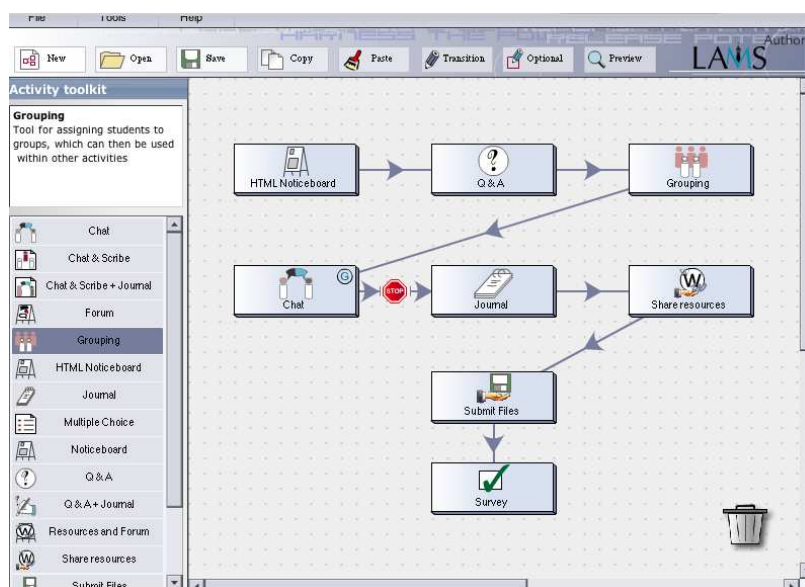


Figura 7 Interfaz de Learning Activity Management System

- **ALFANET** (*Active Learning for Adaptive Internet*) (Figura 8) es una iniciativa europea que tiene como objetivo el desarrollo de nuevos métodos para llevar a cabo un proceso de aprendizaje activo y adaptado al alumno. El

editor de IMS-LD de ALFANET representa los conceptos de IMS-LD mediante una interfaz gráfica.



*Figura 8 Interfaz de Alfabet*

En la Tabla 4 se presenta un resumen de las principales características de las herramientas de autor mencionadas, así como las características de la herramienta mejorada en el presente proyecto de fin de carrera:

Nombre	Desarrollador	Propiedad	Características	Nivel IMS-LD
<b>CopperAuthor</b>	OUNL	Código fuente abierto	<ul style="list-style-type: none"> <li>- Editor de bajo nivel</li> <li>- Interfaz basada en tablas</li> <li>- Herramienta desarrollada en paralelo con el motor de ejecución CopperCore</li> </ul>	Nivel A
<b>RELOAD</b>	JISC (Reino Unido)	Código fuente abierto	<ul style="list-style-type: none"> <li>- Permite la creación y edición de metadatos bajo la especificación IMS-LD y SCORM</li> </ul>	Nivel A Nivel B Nivel C
<b>CoSMoS</b>	Universidad de Duisburg	Código fuente abierto	<ul style="list-style-type: none"> <li>- Destinada inicialmente a dar soporte a la formalización de procesos de aprendizaje colaborativos.</li> <li>- Adaptada posteriormente para dar soporte a los conceptos establecidos por IMS-LD.</li> <li>- Interfaz basada en formularios.</li> </ul>	Nivel A Nivel B Nivel C
<b>MOT+</b>	Centro de investigación LICEF de Canadá	Código fuente abierto	<ul style="list-style-type: none"> <li>- Basada en un modelo orientado a objetos.</li> <li>- Diseñada inicialmente para la estructuración de mapas conceptuales de conocimiento en diferentes dominios.</li> <li>- Extendida posteriormente para dar soporte a IMS-LD aunque no es completamente compatible ya que</li> </ul>	Nivel A



			la herramienta fue desarrollada antes que los estándares de e-learning.	
<b>ASK-LDT</b>	EU Project ICLASS Instituto Informático y Telemático (CERTH) Grecia	Licencia gratuita	<ul style="list-style-type: none"> <li>- Editor de alto nivel que permite definir el flujo entre actividades predefinidas.</li> <li>- No permite la importación de recursos desarrollados con otras aplicaciones.</li> </ul>	Nivel A Nivel B (parcialmente)
<b>LAMS</b>	LAMS Foundation	Código fuente abierto	<ul style="list-style-type: none"> <li>- Permite secuenciar actividades y cursos para diseñar estructuras de aprendizaje de forma sencilla y muy intuitiva</li> <li>- Por sus características no es totalmente compatible con IMS-LD</li> </ul>	Nivel A
<b>ALFANET</b>	aLFanet Project	Código fuente abierto	<ul style="list-style-type: none"> <li>- Editor de bajo nivel</li> <li>- Interfaz web diseñada mediante elementos gráficos</li> </ul>	Nivel A Nivel B Nivel C
<b>CourseEditor</b>	Departamento de Ingeniería Telemática. Universidad Carlos III de Madrid	Licencia gratuita	<ul style="list-style-type: none"> <li>-Permite la creación de cursos que siguen la especificación IMS-LD.</li> <li>- <b>También permite la introducción de características adicionales no contenidas en la especificación que se guardan en documentos independientes.</b></li> <li>- Permite la secuenciación de actividades y tareas de una unidad didáctica en el editor gráfico que posee.</li> </ul>	Nivel A <b>Nivel B</b>

Tabla 4 Tabla comparativa de autoría

## 2.4.2 Motores de ejecución

Los motores de ejecución son herramientas que interpretan el proceso de aprendizaje de una UoL codificada con IMS-LD, monitorizando la realización de las actividades y actualizando el perfil del alumno según los resultados que se vayan obteniendo en las actividades que tienen asignadas. Este tipo de herramientas son utilizadas tanto por profesores como por alumnos a través de una interfaz web adecuada.

- **CopperCore** [20] es un motor de ejecución de UOLs formalizadas con IMS-LD que soporta los tres niveles de la especificación (A, B y C) y que está desarrollado sobre la plataforma Java EE.

Además, la herramienta cuenta con tres interfaces de programación de aplicaciones (API's) que cubren la publicación, administración y entrega del IMS-LD, proporciona una librería de validación e incluye una interfaz de líneas de comandos para las llamadas de la API.

## 2.4.3 Reproductores

Estas herramientas son utilizadas por los actores que interactúan con la UoL, tanto en el proceso de ejecución, como durante el proceso de publicación de la misma.

A continuación se detallan las características de los principales reproductores existentes.

- **CopperCore Player** (Figura 9). Es una versión preconfigurada y lista para usar de CopperCore, en la cual se puede validar e importar cualquier paquete de IMS-LD. Para su utilización es necesario disponer del entorno Java SDK versión 1.4.2\_xx, así como el navegador Microsoft Internet Explorer 6 ó superior.

Cabe destacar que éste ha sido el reproductor que se ha utilizado para realizar las distintas pruebas de secuenciaciones contenidas en el anexo 3 del presente proyecto de fin de carrera.

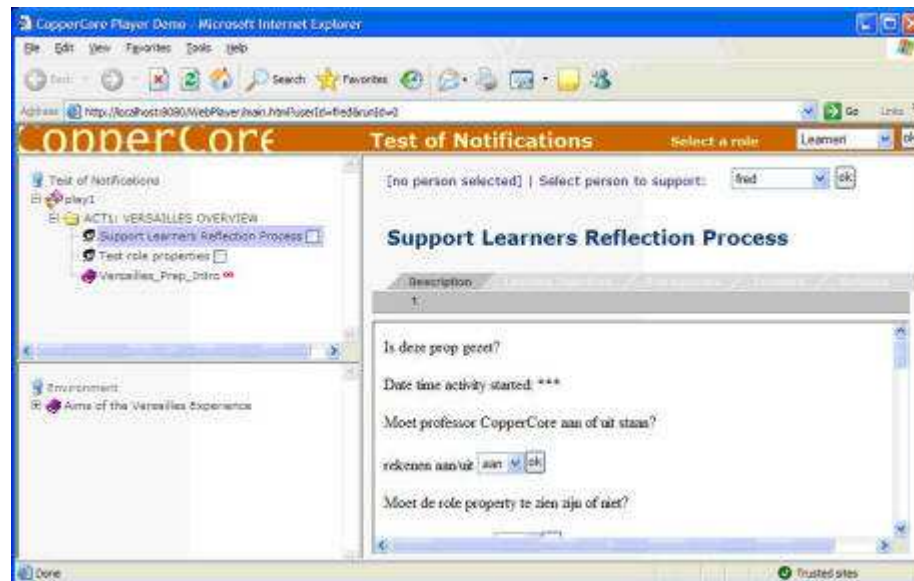


Figura 9 Interfaz de CopperCore player

- **SLeD Player** (Figura 10). SLeD Player es un cliente web desarrollado para el motor de ejecución CopperCore. Forma parte de un proyecto de colaboración entre la *UK Open University* (UKOU) y la *Open University Netherlands* (OUNL). Este reproductor permite la personalización del diseño y de la distribución de la interfaz de reproducción de la UoL mediante el uso de tecnologías XML, y proporciona una interfaz web para la gestión de usuarios y de las ejecuciones de UoLs.

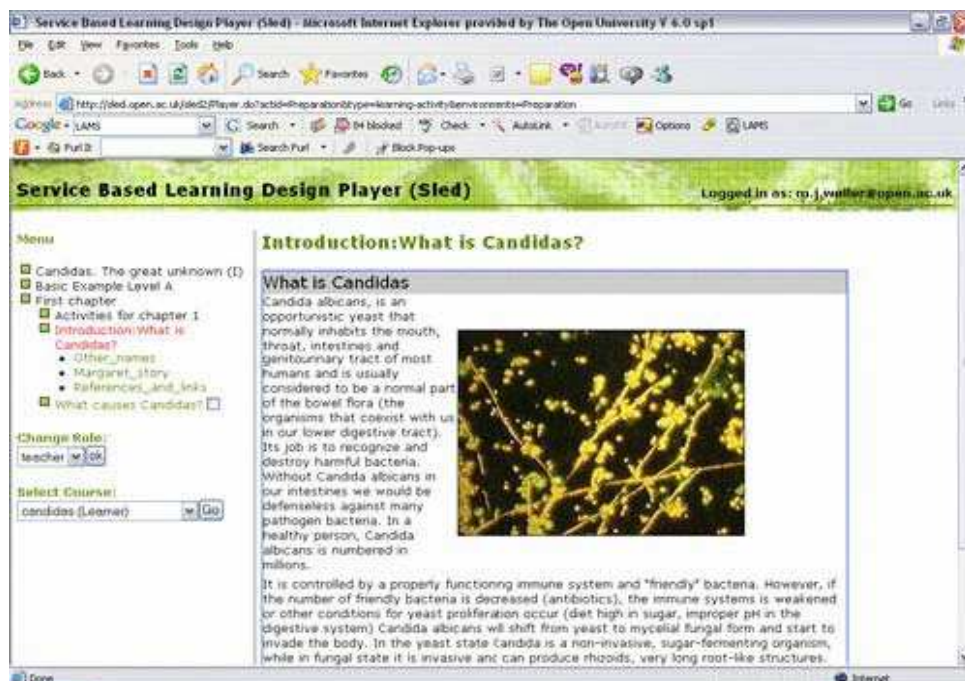


Figura 10 Interfaz de SLeD

- **RELOAD Player** (Figura 11). Es una herramienta construida sobre la plataforma Eclipse, la cual hace uso del CopperCore como motor de ejecución. Fue desarrollada por Paul Sharples y Phillip Beauvoir en la Universidad de Bolton. Está pensada para probar de manera simple las UoLs que se están diseñando con el editor RELOAD.

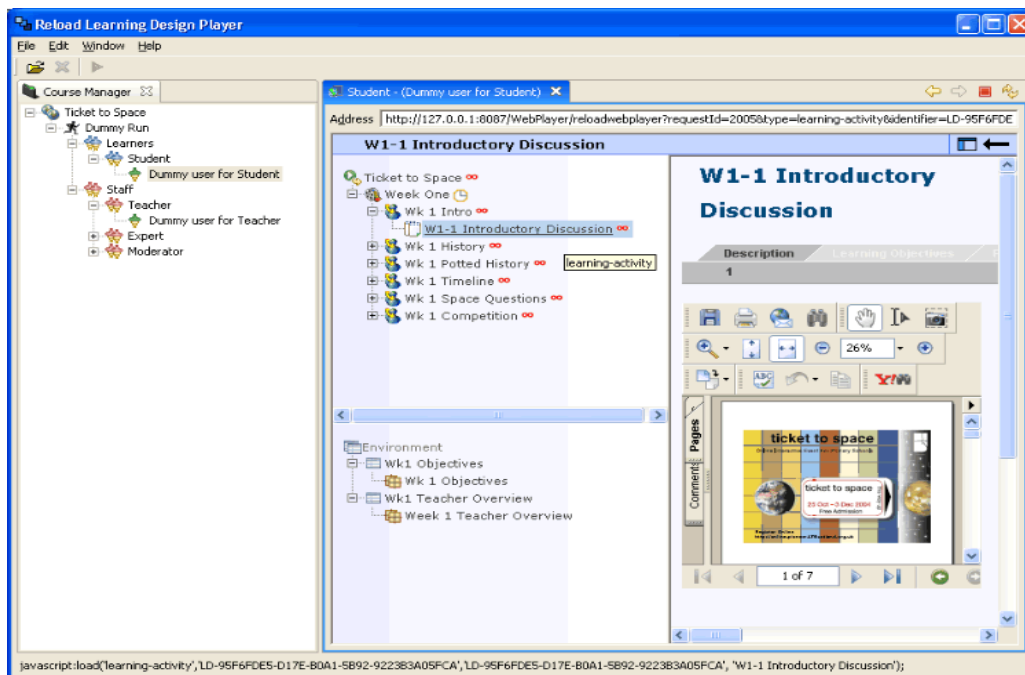


Figura 11 Interfaz de RELOAD Player

- **GRAIL (Gradient-lab RTE for Adaptive IMS-LD in .LRN)** (Figura 12). Es una herramienta desarrollada en la Universidad Carlos III de Madrid que permite la ejecución de UOL en el LMS de código libre .LRN. GRAIL ejecuta OULs de los tres niveles que permite la especificación.

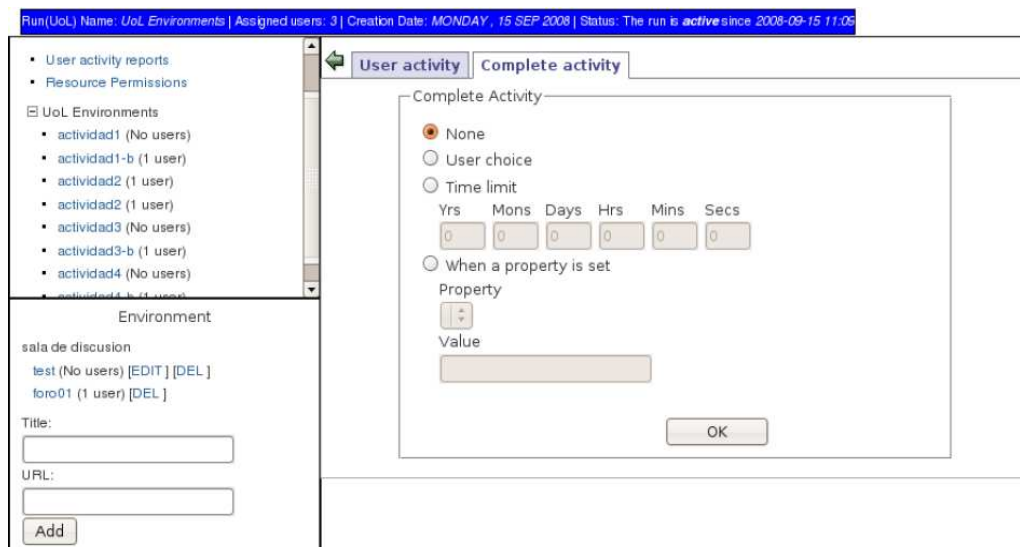


Figura 12 Interfaz de GRAIL

En la Tabla 5 se pueden observar las principales características de los motores de ejecución mencionados:

Nombre	Desarrollador	Propiedad	Características	Nivel IMS-LD
<b>CopperCore</b>	OUNL	Código fuente abierto	<ul style="list-style-type: none"> <li>- Motor de ejecución que soporta la ejecución de UOLs formalizadas con IMS-LD</li> <li>- Cuenta con librerías de validación.</li> </ul>	Nivel A Nivel B Nivel C
<b>CopperCore Player</b>	OUNL	Código fuente abierto	<ul style="list-style-type: none"> <li>- Visor que permite la validación y publicación de paquetes IMS-LD</li> </ul>	Nivel A Nivel B Nivel C
<b>SLeD Player</b>	OU UK y OUNL	Código fuente abierto	<ul style="list-style-type: none"> <li>- Cliente web desarrollado para el motor de ejecución CopperCore</li> </ul>	Nivel A Nivel B Nivel C
<b>RELOAD Player</b>	RELOAD Project	Código fuente abierto	<ul style="list-style-type: none"> <li>- Hace uso de CopperCore como motor de ejecución.</li> <li>- Orientada a probar de manera simple las UoLs que se están diseñando con el editor RELOAD</li> </ul>	Nivel A. Nivel B Nivel C
<b>GRAIL</b>	Universidad Carlos III de Madrid	Código fuente abierto	<ul style="list-style-type: none"> <li>- Permite la ejecución de UOLs en un LMS de código libre (LRN)</li> </ul>	Nivel A Nivel B Nivel C

*Tabla 5 Tabla comparativa de Reproductores*

## 2.5 Tecnologías utilizadas

En este apartado se mostrará una breve descripción de las tecnologías utilizadas y su aplicación al desarrollo del proyecto.

### 2.5.1 Java

Java [21] es un lenguaje de programación de propósito general, concurrente, basado en clases, y orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo. Esto implica que el código que es ejecutado en una plataforma no tiene que ser recompilado para poder ser ejecutado en otra. Java es, a partir del 2012, uno de los lenguajes de programación más populares en uso.

La compañía Sun (la cual fue adquirida por la compañía Oracle) desarrolló la implementación de referencia original para los compiladores de Java, máquinas virtuales, y librerías de clases en 1991 y las publicó por primera vez en el 1995. A partir de mayo del 2007, en cumplimiento con las especificaciones del Proceso de la Comunidad Java, Sun volvió a licenciar la mayoría de sus tecnologías de Java bajo la Licencia Pública General de GNU.

Para la implementación de la aplicación desarrollada en este proyecto de fin de carrera se ha hecho uso de diversas librerías de Java para poder dotar de la funcionalidad deseada a la aplicación.

#### 2.5.1.1 Librerías utilizadas

##### - JCalendar

JCalendar [22] es una librería de código abierto que permite al usuario seleccionar una fecha gráficamente. JCalendar está compuesta de diversos *Java beans*, entre los que se encuentran, JDayChooser, JMonthChooser y JYearChooser. Estos *beans* proporcionan diversos iconos y sus propias propiedades locales de forma que puedan ser usados fácilmente en desarrolladores de GUI. La librería JCalendar también incorpora JDateChooser, el cual está compuesto por un IDateEditor que permite la edición directa de fechas y un botón para abrir el JCalendar y posteriormente seleccionar una fecha.

##### - JGraph

JGraph [23] es un componente Java de código abierto que permite construir editores y visualizadores de grafos en Java. Su principal característica es que se integra perfectamente en Swing, es compatible con las convenciones de código de Java, y puede funcionar tanto en aplicaciones de escritorio como en aplicaciones de servidor. Del resto de sus características destaca su capacidad para gestionar diagramas de múltiples tipos, zoom, visualización en árbol, deshacer o arrastrar y soltar. Puede trabajar con diferentes sistemas, como XML, ficheros, bases de datos, LDAP, etc.

Esta tecnología se utiliza en la aplicación para generar un editor gráfico, en el cual, el usuario decide la secuenciación de las tareas que ha creado previamente.

- **JavaHelp**

JavaHelp [24] es una librería que permite a los desarrolladores la inclusión de un sistema de presentación de la información de ayuda de sus aplicaciones Java de una forma estándar, con todas las funciones y fácil de usar por los usuarios. A la vez, permite al desarrollador personalizar y ampliar el sistema de ayuda para adaptarse al contenido de sus aplicaciones.

El sistema JavaHelp es independiente de la plataforma y funciona en todos los navegadores que soportan la plataforma Java.

- **Xstream**

Xstream [25] es una librería de código abierto que permite serializar objetos a formato XML y viceversa. Entre sus principales características se encuentran su velocidad y bajo consumo de memoria lo que la hace muy adecuada para gráficos de objetos grandes o sistemas de alto rendimiento, la no necesidad de mapeos para la serialización de objetos, la creación de un fichero XML limpio sin información duplicada, lo cual hace más fácil y compacta la lectura del fichero y su compatibilidad total con objetos gráficos.

Esta librería ha sido utilizada para la realización del guardado de los gráficos generado en el editor gráfico para su posterior utilización.

- **JDom**

JDom [26] es una API para manipular documentos XML desde programas implementados en Java.

Sus principales características son su facilidad de uso, su rapidez y su independencia de cualquier analizador. Además se integra bien con SAX y DOM y su adecuamiento a Java lo hace una librería muy útil a la hora de manipular, leer y escribir ficheros en XML

En esta herramienta se utilizará JDOM para:

- Crear el documento IMS-LD que genera la aplicación: Se irán creando uno a uno los elementos XML que especifica el estándar, y se irá añadiendo la información correspondiente a cada uno de ellos.
- Crear los documentos XML adicionales con información que no recoge el estándar.

### - JUnit

JUnit [27] es una biblioteca utilizada para hacer pruebas unitarias de aplicaciones en Java.

Consta de conjunto de clases que permite evaluar si el funcionamiento de cada uno de los métodos de las clases Java se comporta como se espera. Es decir, en función de algún valor de entrada se evalúa el valor de retorno esperado. De esta forma, si la clase cumple con la especificación, entonces JUnit devolverá que el método de la clase pasó exitosamente la prueba, por el contrario, en caso de que el valor esperado sea diferente al que regresó el método durante la ejecución, JUnit devolverá un fallo en el método correspondiente.

Esta librería ha sido utilizada en este proyecto de fin de carrera para la realización de la evaluación de la herramienta resultante después de haber introducido todos los cambios de interfaz y nueva funcionalidad requerida.

### 2.5.2 XML

XML, siglas en inglés de *eXtensible Markup Language*, es un lenguaje de marcado [28]. El W3C (World Wide Consortium) es el responsable del desarrollo de la especificación, la versión 1.0 de XML fue aceptada el 10 de febrero de 1998. Deriva del lenguaje SGML y permite definir la gramática de lenguajes específicos para estructurar documentos grandes.

XML ha sido una de las bases de la difusión y popularidad de la Web. No obstante, XML no sólo ha nacido para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo, etc.

Entre sus principales ventajas se encuentran que el lenguaje es extensible, ya que permite incorporar tus propios tags; es interoperable, pues no depende de sistema operativo, lenguaje o fuente de datos de aplicaciones que lo gestionen; y, por último, es autodescriptivo, su estructura es fácilmente identificable en aplicaciones futuras.

IMS define una serie de etiquetas para representar los diferentes elementos que deben especificarse al diseñar un proceso de enseñanza-aprendizaje, como los objetivos didácticos del mismo (*<learning-objectives>*), los participantes (*<roles>*), las actividades a realizar (*<activities>*), el método pedagógico a aplicar (*<method>*), los objetos didácticos a utilizar (*<learning-object>*), o condiciones (*<conditions>*) para definir las condiciones de nivel B [29].

Por lo tanto, toda la información que el usuario rellena en la aplicación, se almacena en formato XML, que es el que requiere la especificación IMS-LD.

A continuación se muestra un ejemplo de archivo XML que utiliza las etiquetas definidas por IMS para representar los diferentes elementos que deben especificarse al diseñar un proceso de enseñanza aprendizaje.



```

<learning-design>
  <title>Curso de Introducción a la Informática</title>
  <learning-objectives>Manejar el ordenador libremente</learning-
objectives>
  <prerequisites>No es necesario ningún conocimiento
previo</prerequisites>
  <components>
    <roles>
      <learner>Alumno</learner>
      <staff>Profesor </staff>
    </roles>
    <activities>
      <support-activity>Resolver las dudas del
alumno</support-activity>
      <activity-structure>
        <learning-activity>Crear una carpeta
nueva</learning-activity>
        <learning-activity>Eliminar una carpeta
existente</learning-activity>
      </activity-structure>
    </activities>
    <environments>
      <environment>
        <learning-object>Documento.doc</learning-
object>
      </environment>
      <environment>
        <service><conference>Debatir las dificultades
observadas</conference></service>
      </environment>
    </environments>
  </components>
  <method>
    <play>
      <act>
        <role-part>Alumno</role -part>
        <complete-act>Fin de la unidad
didáctica</complete-act>
      </act>
      <complete-play>Fin del curso</complete-play>
    </play>
  </method>
</learning-design>

```

### 2.5.3 Eclipse

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse y que son usados también para desarrollar el mismo Eclipse.

Eclipse fue desarrollado originalmente por IBM Canada como el sucesor de su familia de herramientas para VisualAge. Actualmente es desarrollado por la Fundación Eclipse [30], una organización independiente sin ánimo de lucro que fomenta una comunidad de Código abierto y un conjunto de productos complementarios, capacidades y servicios. En noviembre del 2001, se formó un consorcio para el desarrollo futuro de Eclipse como Código abierto.

Algunas de las ventajas que presenta la herramienta Eclipse son:

- Emplea módulos para proporcionar toda su funcionalidad a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no.
- Este mecanismo de módulos es una plataforma ligera para componentes de software. Adicionalmente permite a Eclipse extenderse usando otros lenguajes de programación como son C/C++ y Python, permite a Eclipse trabajar con lenguajes para procesado de texto como LaTeX, aplicaciones en red como Telnet y Sistema de gestión de base de datos.
- La arquitectura de módulos permite escribir cualquier extensión deseada en el ambiente, como sería Gestión de la configuración. Se provee soporte para Java y CVS en el SDK de Eclipse.

En la realización de este proyecto de fin de carrera, el uso de Eclipse ha permitido la instalación de diferentes plugins que han facilitado la implementación de la herramienta como por ejemplo un editor de construcción de ventanas o para la evaluación de la misma, como por ejemplo el entorno de pruebas JUnit.

### 2.5.4 Bugzilla

Bugzilla [31] es un servidor que permite a los desarrolladores de aplicaciones llevar un seguimiento del producto desarrollado de forma efectiva.

Bugzilla es una herramienta que permite la introducción de *bugs* y cambios en el código, la introducción y revisión de parches, así como la comunicación con compañeros de equipo en caso de que el producto sea desarrollado entre varias personas.

Para la realización de este proyecto se ha utilizado la herramienta de propósito general Bugzilla instalada y configurada para uso propio por el laboratorio GAST de la Universidad Carlos III. [32] En ella se han ido introduciendo todas las mejoras que eran necesarias en la herramienta CourseEditor en forma de *bugs* y se ha llevado un seguimiento del grado de realización de las mismas.



# Capítulo 3

## Desarrollo del proyecto

En este capítulo se presenta una descripción detallada de todas las funcionalidades añadidas y los cambios realizados a la herramienta CourseEditor y se explican las características fundamentales de cada uno de los módulos así como su funcionalidad.

### 3.1 Introducción

Este proyecto de fin de carrera tiene como objeto la realización de los cambios y mejoras necesarios en la herramienta CourseEditor previamente desarrollada, así como la introducción de nuevas funcionalidades que permitan al usuario la creación de cursos completos de una manera simple y compatible con IMS-LD sin que el usuario tenga conocimientos sobre dicho estándar.

Dichas mejoras se realizarán con el fin de hacer la navegación por la herramienta mucho más útil y usable de forma que al usuario no le resulte tan tediosa la tarea de definir las características de su curso. Para ello, se añadirán nuevos botones, y se facilitará el guardado de la información de cada una de las pestañas, entre otros. En este apartado se explica detalladamente todas las mejoras introducidas así como su finalidad.

Para la presentación de las nuevas características introducidas en CourseEditor, se ha definido un curso basado en la asignatura ‘Programación de Sistemas’ de grado en Ingeniería Telemática, cuyo diagrama de planificación temporal puede encontrarse en la siguiente URL: [http://www3.uc3m.es/reina/CRONOGRAMAS/Idioma\\_1/2013/](http://www3.uc3m.es/reina/CRONOGRAMAS/Idioma_1/2013/)

215.13409.pdf. Además, en el anexo 2 de la presente memoria se puede encontrar una descripción más detallada del mismo.

La asignatura consta de 6 créditos y se imparte en 29 sesiones ó unidades didácticas, consistentes en 15 sesiones de teoría y 14 sesiones de prácticas que se distribuyen a lo largo de 14 semanas. Como prerrequisito se requiere que el alumno haya cursado previamente la asignatura de ‘Programación’.

Mediante la realización de la asignatura, se pretende que el alumno alcance una serie de objetivos, como por ejemplo interpretar, explicar y ejemplificar diferentes conceptos de la programación orientada a objetos, tales como recursión, ó estructuras de datos y la implementación de interfaces de usuario sencillas.

Para la realización de esta asignatura el alumno contará con distintos tipo de tareas a realizar tales como prácticas, exámenes parciales, foros, chats ó proyectos.

Para la impartición del curso, serán necesarios distintos perfiles de personas que llevarán a cabo funciones diferentes a lo largo de la realización de curso, también denominados roles. Para esta asignatura en concreto existirán un profesor de teoría, un profesor de prácticas y los alumnos que cursará la asignatura.

Finalmente, dentro de cada sesión el alumno realizará una serie de tareas, en un determinado orden con el fin de llevar a cabo con éxito el proceso de aprendizaje. Cada tarea deberá pertenecer a un tipo de tarea y a una unidad didáctica, además debe tener asignado al menos un rol que se encargue de llevar a cabo dicha tarea.

La definición completa del curso definido se puede encontrar en el anexo 2 del presente proyecto de fin de carrera.

Como se puede observar, el diseño de un curso requiere introducir una gran cantidad de información. Debido a su especificidad, parte de esta información no esta recogida dentro del estándar IMS-LD. CourseEditor permite recoger toda esta información en un mismo diseño pero manteniendo la parte compatible con IMS-LD de la información adicional en ficheros XML separados que recogen la información general del curso, las unidades didácticas, los tipos de tareas con sus correspondientes campos y las tareas generadas.

Así mismo, la herramienta posee un editor gráfico que permite al usuario secuenciar las actividades que se realizarán en cada unidad didáctica. Por lo tanto, si el usuario hace uso de dicho editor se almacenará un fichero XML por cada unidad didáctica que contendrá información de las tareas, las relaciones entre ellas y las condiciones que permitirán pasar de una a otra. Toda esta información permitirá la reconstrucción del gráfico en una ejecución posterior de la herramienta.

Las mejoras introducidas se dividen en cuatro bloques fundamentales:

- Cambios de interfaz de la herramienta.
- Errores de la herramienta.

- Nuevas funcionalidades.
- Evaluación de la herramienta.

## 3.2 Cambios en la interfaz de la herramienta

### 3.2.1 Introducción de un Menú

Tal y como se puede observar en la Figura 13, en la versión anterior de la herramienta, todas las funcionalidades básicas tales como guardar, abrir o crear un nuevo curso, así como la visualización de la ayuda se ejecutaban desde diferentes botones situados en la parte inferior de la interfaz. Esto resultaba poco intuitivo para un usuario con conocimientos básicos de informática puesto que las opciones “Guardar”, “Abrir” y “Nuevo” son opciones habituales en muchas herramientas ofimáticas que habitualmente están asociadas a otros elementos del interfaz (típicamente menús).

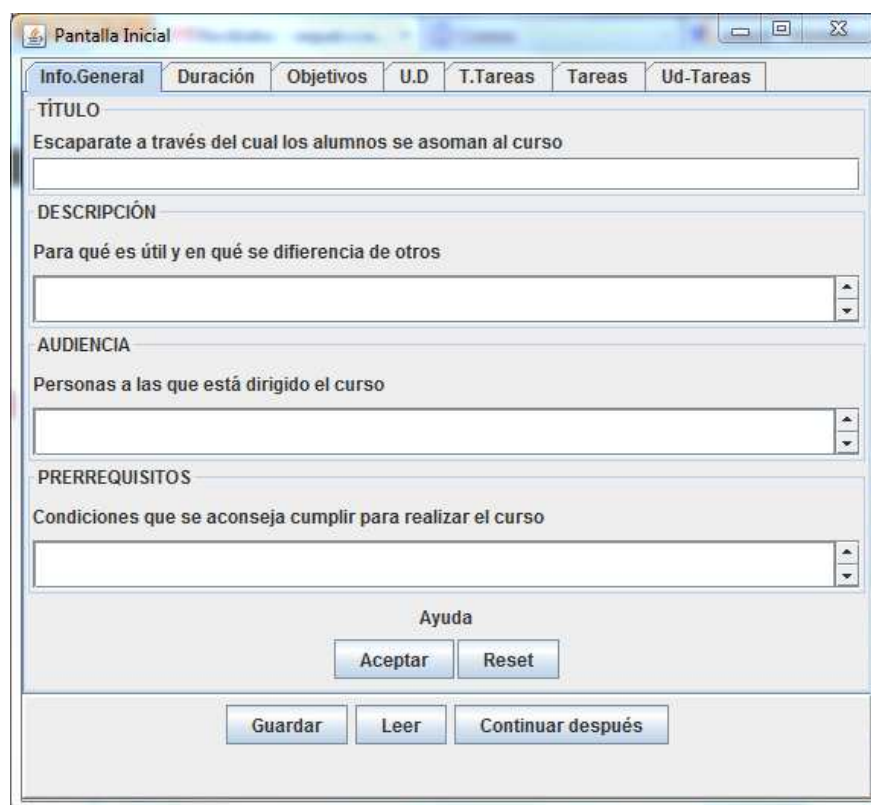


Figura 13 Pestaña Info. General de la herramienta CourseEditor v1.0

Con el objeto de minimizar el número de botones, así como de hacer el uso de estas funcionalidades más sencillo y parecido a otras herramientas de uso común entre profesores como paquetes ofimáticos o navegadores de internet, se ha implementado un menú desplegable en la parte superior de la interfaz, el cual recoge todas las funcionalidades mencionadas.

## CAPÍTULO: Desarrollo del proyecto

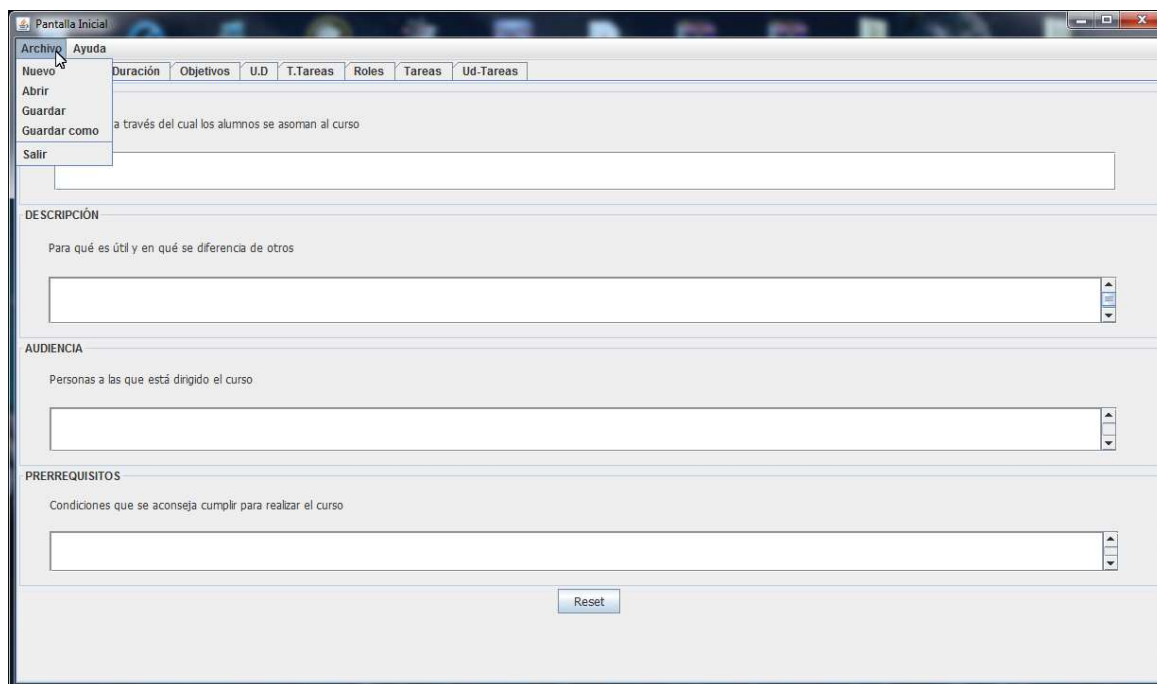


Figura 14 Pestaña Info. General de la herramienta CourseEditor v2.0

En la Figura 15 se puede ver con más detalle el menú implementado.

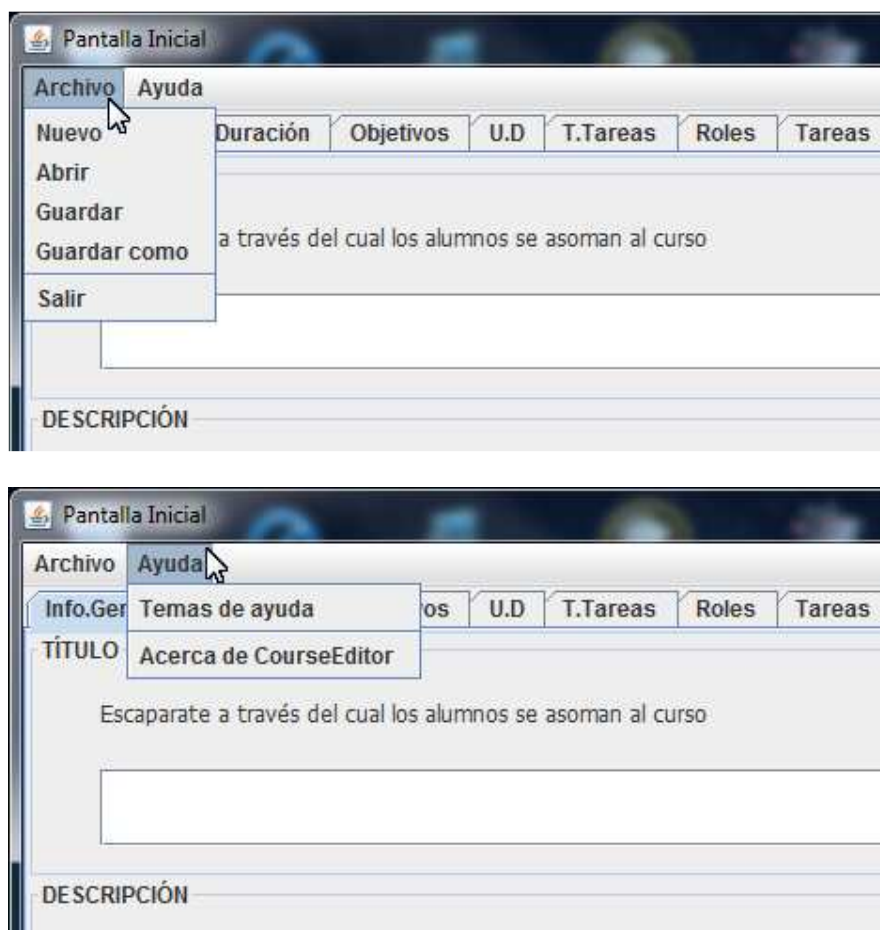


Figura 15 Detalle de las funcionalidades del menú de CourseEditor v2.0



El uso de menús en una aplicación software tiene ventajas y desventajas.

Su principal ventaja es la organización y agrupación de la información. De esta forma se logra tener la mayor parte de la funcionalidad de la herramienta agrupada, mostrando todas las opciones disponibles para el usuario, de forma que es más fácil encontrarlas, ya que en la versión anterior, esta funcionalidad se encontraba repartida en botones situados en distintas partes de la interfaz de usuario.

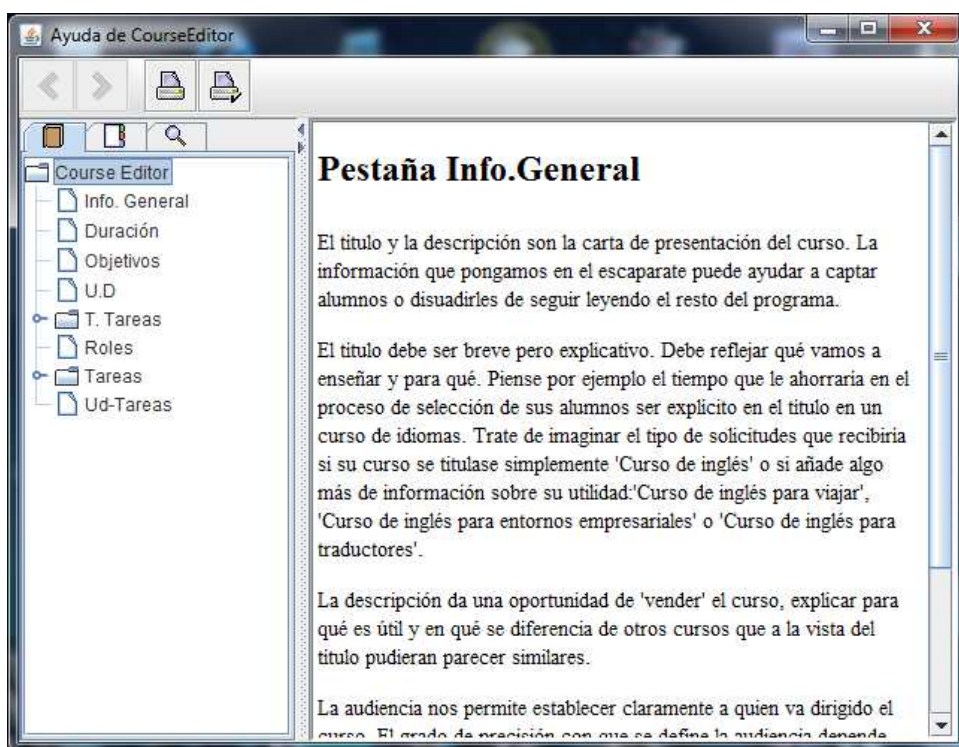
Por contra, el uso de menús puede ser lento para usuarios expertos. No obstante, para el desarrollo de la herramienta se ha partido con la premisa de que el usuario puede tener poca o nada de experiencia tanto en el conocimiento de la especificación IMS LD como en el uso de herramientas software, por lo tanto el uso de un menú para este tipo de usuarios es altamente recomendado.

Una vez valorados tanto las ventajas como los inconvenientes, se ha llegado a la conclusión de que el uso de un menú desplegable introduce una gran mejora a la herramienta, ya no solo visualmente, sino también de facilidad de uso.

### **3.2.2 Acceso a la ayuda de la herramienta**

La versión anterior de la herramienta permitía acceder a la ayuda de cada pestaña, desde la etiqueta pulsable de Ayuda situada en cada una de las pestañas y ventanas. Su inconveniente principal era que para visualizar el contenido de toda la ayuda, el usuario debía irse desplazando por cada una de las pestañas para pulsar la etiqueta correspondiente.

En esta nueva versión, la ayuda se ha integrado en el menú, y utilizando la librería JavaHelp se ha implementado una ventana de ayuda desde la cual el usuario puede navegar por la ayuda de toda la aplicación así como realizar búsquedas por palabra o imprimir la información ofrecida. La Figura 16 muestra la nueva ventana de ayuda de la aplicación.



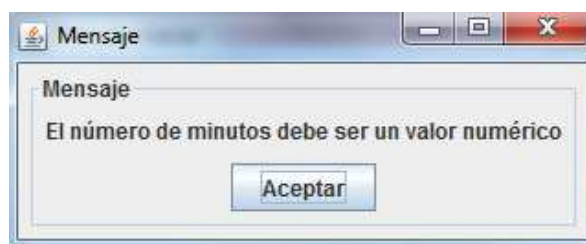
*Figura 16 Ventana de ayuda de CourseEditor v2.0*

El uso de JavaHelp ofrece una gran cantidad de ventajas entre las que cabe destacar: [33]

- JavaHelp permite que la información pueda ser actualizada dinámicamente o extendida mediante funcionalidad adicional sobre web o intranet.
- Permite la agrupación de toda la información.
- La librería está escrita íntegramente en Java, lo cual la hace adecuada para cualquier tipo de plataforma que soporte Java Runtime Environment (JRE).
- Es una interfaz familiar para el usuario, la cual consta de tabla de contenidos, índice, búsqueda en todo el texto y navegación.

Debido a todas estas características se ha decidido mejorar la implementación de la ayuda de la herramienta.

Por otra parte, se proporciona una ayuda contextual en la herramienta, de forma que si el creador de cursos introduce una información errónea que requiere introducirse correctamente en el acto, aparece una ventana en la que se informa al usuario del error producido. En la Figura 17 se puede observar un ejemplo de la ayuda contextual proporcionada.



*Figura 17 Ayuda contextual de CourseEditor v2.0*

### 3.2.3 Selección de la ubicación de guardado de ficheros

En la versión anterior, toda la documentación generada por la herramienta (el fichero de texto que contiene la información necesaria para reconstruir un curso cargándolo desde fichero en la herramienta y los ficheros XML en los que se incluyen, el fichero compatible con la especificación IMS-LD, tres ficheros de información adicional no recogida en la especificación y las secuenciaciones de las unidades didácticas generadas) se guardaban automáticamente en la misma ubicación donde se encontraba la herramienta.

En esta nueva versión se ha optado por implementarlo de tal forma que al seleccionar la opción ‘Guardar’ ó ‘Guardar como’ del menú, se abra una ventana que nos permita elegir la ubicación donde guardaremos la carpeta que contenga todos los documentos generados.

Siempre que el usuario seleccione la opción ‘Guardar como’ se pedirá que seleccione una ubicación y dé un nombre al curso creado.

Cuando el usuario seleccione la opción ‘Guardar’, la primera vez que se pulse, aparecerá la ventana de selección de ubicación e introducción de nombre para el curso, y las siguientes se sobrescribirá automáticamente con la nueva información introducida.

Además, la carpeta que contiene la documentación generada se guardará comprimida en formato zip. De esta forma, se consigue un fichero que será compatible con muchas de las herramientas que siguen la especificación IMS-LD, como por ejemplo la herramienta en la que se puede realizar simulaciones de los cursos creados, CopperCore Player.

La principal ventaja que aporta esta funcionalidad es la de evitar confusión con respecto a si se ha guardado bien la información del curso generado y la ubicación exacta del mismo.

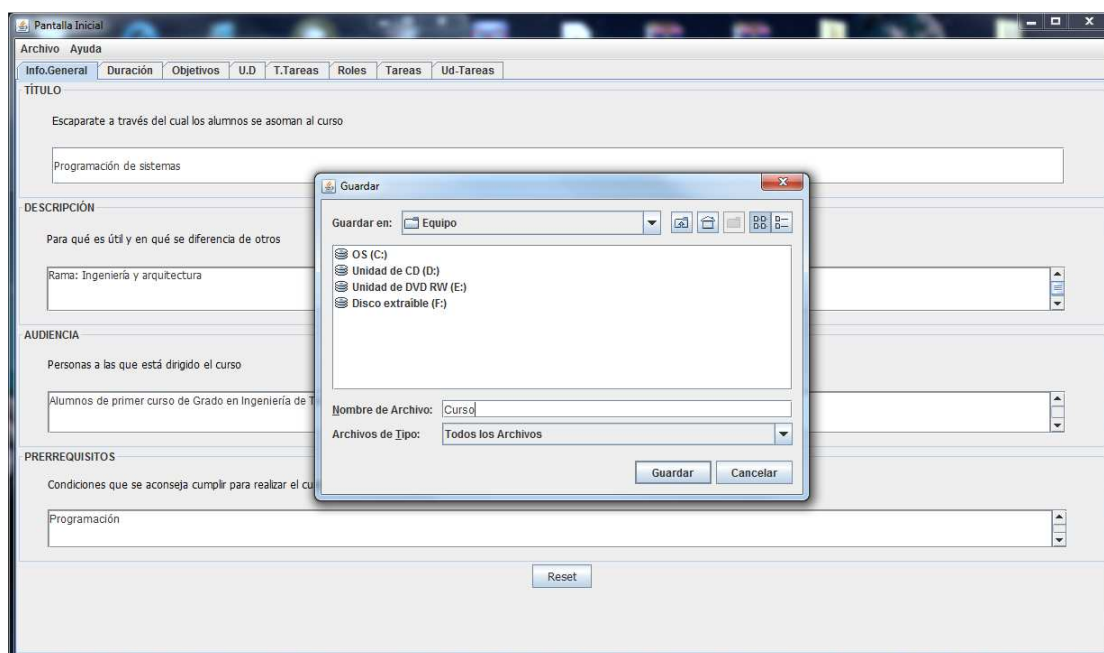


Figura 18 Selección de la ubicación de guardado en CourseEditor v2.0

Si el nombre del archivo que deseamos guardar ya existe, aparecerá una ventana preguntando al usuario si desea sobrescribir la información. Si seleccionamos la opción ‘Si’, la información se sobrescribirá.

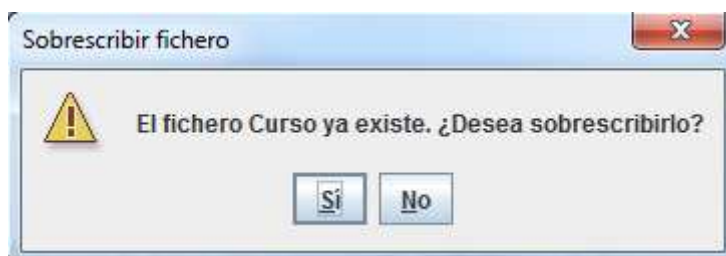


Figura 19 Ventana emergente que pregunta si se desea sobrescribir en CourseEditor v2.0

### 3.2.4 Guardado automático de la información

En la versión anterior de la herramienta, el usuario debía pulsar el botón “Aceptar” cada vez que se cambiaba de pestaña. Esto permitía ir guardando la información introducida en cada una de las pestañas y que ésta se tuviera en cuenta a la hora de rellenar las siguientes.

The screenshot shows a software window titled 'Pantalla Inicial' with a standard Windows-style title bar. Inside the window, there is a tabbed interface with six tabs: 'Info.General' (selected), 'Duración', 'Objetivos', 'U.D', 'T.Tareas', and 'Ud-Tareas'. The 'Info.General' tab contains four text input fields, each with a label and a description:

- TÍTULO**: Escaparate a través del cual los alumnos se asoman al curso. Below it is a single-line text input field.
- DESCRIPCIÓN**: Para qué es útil y en qué se diferencia de otros. Below it is a multi-line text input field.
- AUDIENCIA**: Personas a las que está dirigido el curso. Below it is a single-line text input field.
- PRERREQUISITOS**: Condiciones que se aconseja cumplir para realizar el curso. Below it is a multi-line text input field.

Below the input fields, there is a section labeled 'Ayuda' containing two buttons: 'Aceptar' and 'Reset'. At the bottom of the window, there are three buttons: 'Guardar', 'Leer', and 'Continuar después'.

*Figura 20 Guardado de la información de cada pestaña en CourseEditor v1.0*

Esto se ha mejorado eliminando el botón “Aceptar” de cada una de las pestañas de la interfaz y realizando el guardado de la información automáticamente y de forma totalmente transparente para el usuario.

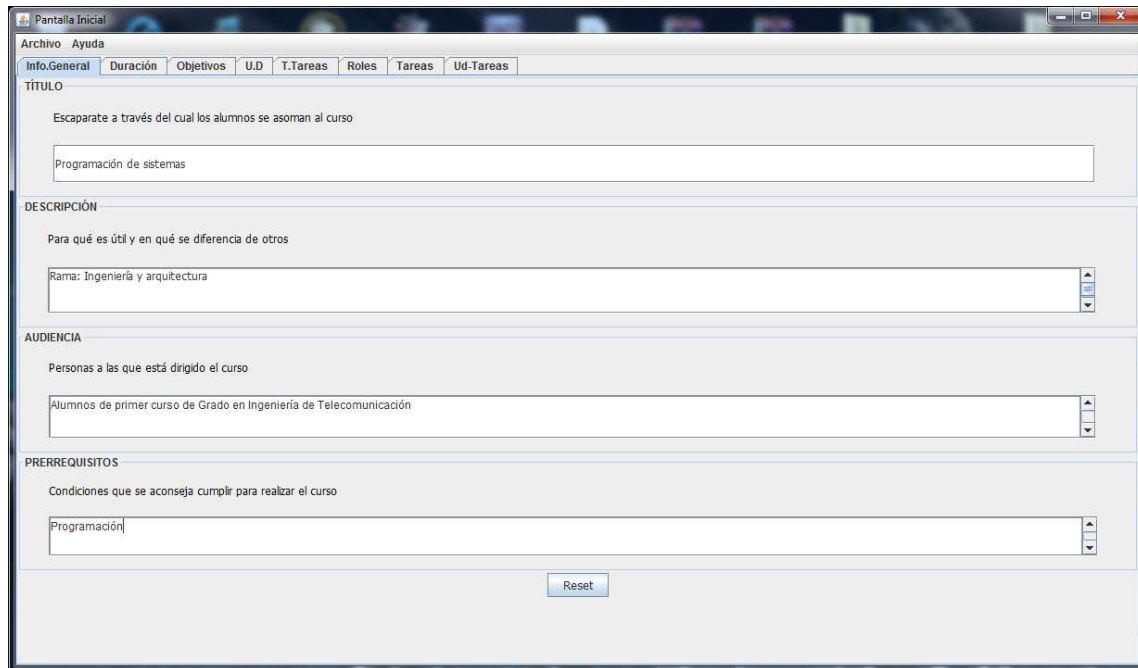


Figura 21 Guardado de la información de cada pestaña en CourseEditor v2.0

Por otra parte, también se ha implementado un mensaje de seguridad de forma que cuando se pulsa el botón aspa [X] de la ventana para cerrar la aplicación, sale una ventana que pregunta si realmente se desea salir sin haber guardado la información del curso introducida.

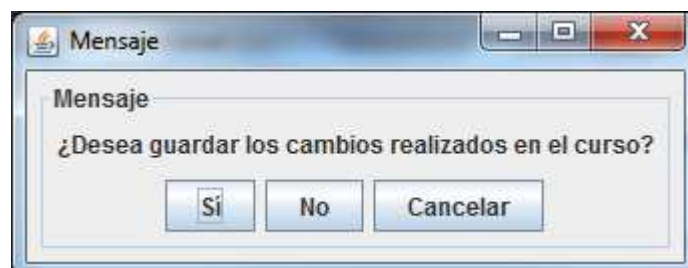


Figura 22 Mensaje de cierre de la aplicación en CourseEditor v2.0

De ésta forma, si el usuario cierra por error la aplicación antes de haber guardado la información del curso introducida tiene un mecanismo de seguridad adicional que le obliga a confirmar dicha acción.

### 3.2.5 Detección de errores producidos en la introducción de la información de un curso

Si se ha cometido algún error durante la introducción de información en todas las pestañas de la herramienta, o alguna información obligatoria no se ha rellenado aún, cuando el usuario pulse en las opciones 'Guardar' ó 'Guardar como' aparecerá una ventana en la que se lista todos los errores que debe subsanar el usuario con el fin de crear un curso correcto. El usuario tiene la opción de pulsar el botón 'Cancelar' de la

ventana y volver a la aplicación a completar la información de forma correcta, ó guardar de todos modos el curso incompleto si se pulsa el botón 'Aceptar'.



*Figura 23 Mensaje que lista los errores cometidos en la introducción de datos en CourseEditor v2.0*

Para implementar esta funcionalidad se ha creado un vector de errores, en el que se guardan todos los errores que se van cometiendo cuando se va guardando la información. Este vector con los mensajes es el que será mostrado en el mensaje.

Si no se ha producido ningún error en la introducción de datos, al seleccionar la opción 'Guardar' o 'Guardar como', este mensaje no aparecerá y únicamente se guardara la información de la pestaña.

Por otro lado, y paralelamente a esto, todos los errores que se hayan producido y por tanto se hayan listado en la ventana de errores serán marcados en rojo en todas las pestañas de la herramienta de forma que es mucho más fácil para el usuario localizar los errores cometidos.

En las Figuras 24 y 25 podemos ver un par de ejemplos de esta mejora:



Pantalla Inicial

Archivo Ayuda

Info.General Duración Objetivos U.D T.Tareas Roles Tareas Ud-Tareas

TÍTULO

Escaparate a través del cual los alumnos se asoman al curso

Programación

DESCRIPCIÓN

Para qué es útil y en qué se diferencia de otros

AUDIENCIA

Personas a las que está dirigido el curso

PRERREQUISITOS

Condiciones que se aconseja cumplir para realizar el curso

Reset

*Figura 24 Señalización en la pestaña Info.General de los errores cometidos al introducir la información en CourseEditor v2.0*

Pantalla Inicial

Archivo Ayuda

Info.General Duración Objetivos U.D T.Tareas Roles Tareas Ud-Tareas

DURACIÓN

Duración en horas del curso:

Número de horas: 0 horas

Fecha Inicio 11/06/2013

Fecha Fin 11/06/2013

CRÉDITOS ECTS

1 crédito ECTS equivale a 25 o 30 h de trabajo del alumno tanto en el aula como de trabajo personal en casa

Número de créditos: 6 créditos

Reset

*Figura 25 Señalización en la pestaña Duración de los errores cometidos al introducir la información en CourseEditor v2.0*

### 3.2.6 Cambios en la pestaña 'Duración'

La pestaña 'Duración' permite introducir toda la información relacionada con la duración del curso: fecha de inicio y de fin, así como el número de horas del curso y el número de créditos ECTS.



En esta pestaña se han llevado a cabo los siguientes cambios:

- Se ha añadido el botón “Reset” que permite eliminar con una sola acción el contenido de todos los campos.
- Para la introducción de fechas, antes se ofrecía al usuario un *jcomboBox* en el que podía seleccionar unas fechas limitadas y restringidas a unos determinados años. En la nueva versión se ha añadido una nueva funcionalidad utilizando la librería JCalendar que permite añadir un calendario a la aplicación y que permite al usuario escoger la fecha deseada sin limitación.

También, se ha eliminado la restricción de tener que seleccionar el número de horas y el de número de créditos de un conjunto predeterminado. Para ello se han cambiado los *jcomboBox* que había, por campos de introducción de texto (*TextField*), y se han llevado a cabo las comprobaciones necesarias para que la introducción se realice de forma correcta, es decir, que el usuario no pueda introducir caracteres no permitidos como por ejemplo, caracteres de texto.

En la Figura 26 se puede observar la existencia de restricciones en la selección de fechas y de números.

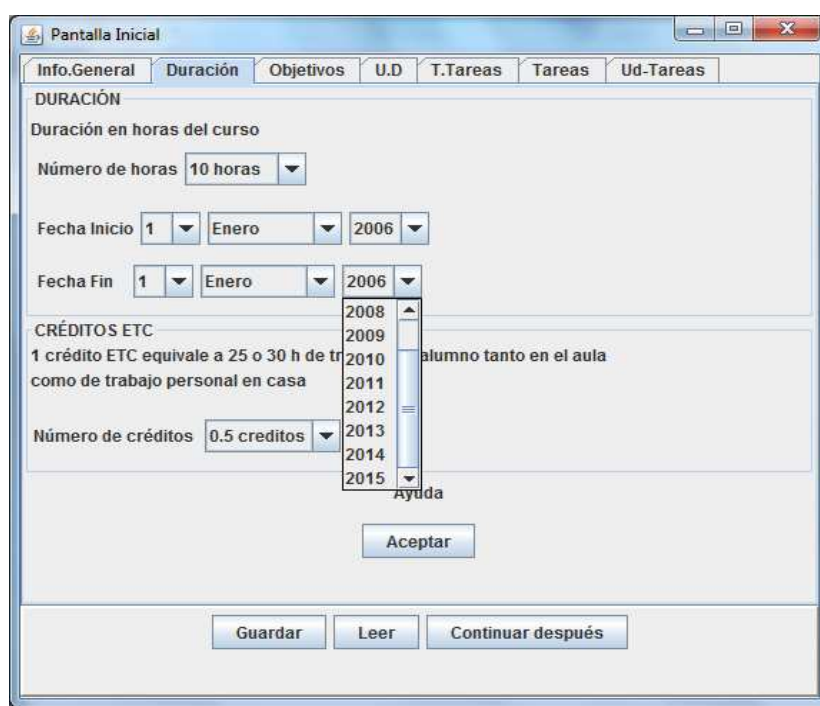


Figura 26 Pestaña Duración de CourseEditor v1.0

En la Figura 27 y Figura 28 se puede observar los cambios mencionados:

Figura 27 Pestaña Duración de CourseEditor v2.0

Figura 28 Calendario y campos de texto libres añadidos de CourseEditor v2.0

### 3.2.7 Cambios en la pestaña 'Objetivos'

Todo curso debe definir y describir en detalle lo que se espera conseguir después de haber participado en él. De esta forma, tiene que tener definidos una serie de objetivos

o metas a alcanzar por el alumno, los cuales tendrán mayor o menor consideración en la evaluación dependiendo de su importancia.

En esta pestaña el usuario puede introducir los objetivos del curso. Estos tienen dos campos fundamentales: *Nombre* y *Peso*. El *Nombre* corresponderá con el título asignado a dicho objetivo y el *Peso* define la importancia relativa dentro del curso. Este peso está normalizado y la suma total de pesos de todos los objetivos del curso no puede ser superior a 1.

Se pueden crear varios objetivos para un curso y cada uno de estos objetivos, con su correspondiente peso, se almacena en una tabla de objetivos.

Los cambios llevados a cabo en esta pestaña son:

- En la versión 1.0 de la herramienta aparecía un botón de “Borrar” para la tabla que servía para borrar un objetivo previamente seleccionado en la tabla. Este botón se ha sustituido por un botón con icono con forma de aspa para cada ítem que permite eliminar los objetivos de manera independiente con un solo click.

De esta forma, al usuario le resultara más visual y, al mismo tiempo menos confuso utilizar las opciones, eliminando así la necesidad de que el usuario tenga que, primero seleccionar el objetivo en concreto que desea eliminar, y después buscar el botón correspondiente a la acción deseada.

- En la versión 1.0, no existía la posibilidad de editar los objetivos si el usuario deseaba cambiarlos. Por lo tanto se ha añadido también un botón “Editar” con un icono de lápiz para cada ítem. Al pulsar este botón se carga la información relativa al objetivo seleccionado en los espacios de introducción de información para que el usuario modifique lo que considere oportuno. Una vez modificada la información se volverá a pulsar el botón ‘Añadir’ para añadir la nueva información a la tabla.
- Se ha eliminado la selección de tipo de objetivos (genérico y específico)
- Se indica y comprueba que el peso de todos los objetivos debe sumar 1.

En las Figuras 29 y 30 se muestran los cambios realizados.

**Objetivos**

Definición y peso de los objetivos a alcanzar en el curso

Nombre:  Peso:  Tipo: Genérico ▼ Añadir

Objetivo	Peso	Tipo	Conceptu...	Procedim...	Actitudinal	Otros

Peso total objetivos:

Borrar

**Ayuda**

Aceptar Reset

Guardar Leer Continuar después

Figura 29 Pestaña Objetivos de CourseEditor v1.0

**Objetivos**

Definición y peso de los objetivos a alcanzar en el curso

Nombre:  Peso:  Añadir

Objetivo	Peso	Conceptual	Procedimental	Actitudinal	Otros	Editar	Borrar
Objetivo 1: Interpretar, explicar y ejemplificar los mecanismos ...	0.125	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Objetivo 2: Interpretar, explicar y ejemplificar los conceptos de...	0.125	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Objetivo 3: Interpretar, explicar y ejemplificar los fundamentos ...	0.125	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Objetivo 4: Interpretar, explicar y ejemplificar los distintos algo...	0.125	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Objetivo 5: Interpretar, explicar y ejemplificar los fundamentos...	0.125	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Objetivo 6: Ser capaz de crear, compilar y ejecutar un program...	0.125	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Objetivo 7: Explicar y presentar de forma concisa el trabajo re...	0.125	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Objetivo 8: Ejemplificar y aplicar estrategias para resolver los ...	0.09	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Peso total objetivos:

Reset

Figura 30 Pestaña Objetivos de CourseEditor v2.0

En la Figura 31 se puede observar con más detalle los botones de editar y borrar.

The screenshot shows a web interface for CourseEditor v2.0. At the top, there is a 'Peso' label followed by an empty text input box and a blue 'Añadir' button. Below this is a table with six columns: 'Conceptual', 'Procedimental', 'Actitudinal', 'Otros', 'Editar', and 'Borrar'. Each of the first four columns contains a row of eight small square checkboxes. The 'Editar' column contains eight green pencil icons, and the 'Borrar' column contains eight red 'X' icons. The table is set against a light gray background.

*Figura 31 Botones Editar y Borrar de la tabla de CourseEditor v2.0*

Tal y como se observa en la Figura 31, los objetivos se pueden clasificar en uno, o más de uno, de los siguientes tipos:

- Conceptuales: objetivos centrados en la asimilación y comprensión de conceptos.
  - Procedimentales: objetivos centrados en la adquisición de habilidades y destrezas.
  - Actitudinales: objetivos centrados en la formación de hábitos y actitudes correctas en relación a los temas tratados.
  - Otros: objetivos que tienen otro fin distinto a los anteriores.
- Por otra parte, como las descripciones de cada objetivo normalmente son bastante largas, no se puede visualizar la información completa en la tabla, por ello se ha implementado de forma que cuando se pase el ratón por encima de la descripción deseada, esta se muestra completa en forma de *tool tip* ó descripción emergente. [34]

En la Figura 32 se puede observar la descripción emergente implementada.

Objetivo	Peso	Conceptual	Procedimental	Actitudinal	Otros	Editar	Borrar
Objetivo 1: Interpretar, explicar y ejemplificar los mecanismos ...	0.125	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Objetivo 2: Interpretar, explicar y ejemplificar los conceptos de...	0.125	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Objetivo 3: Interpretar, explicar y ejemplificar los fundamentos...	0.125	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Objetivo 4: Interpretar, explicar y ejemplificar los fundamentos de recursión y complejidad. Representar, diseñar e implementar algoritmos recursivos.	0.125	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Objetivo 5: Interpretar, explicar y ejemplificar los fundamentos...	0.125	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Objetivo 6: Ser capaz de crear, compilar y ejecutar un program...	0.125	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Objetivo 7: Explicar y presentar de forma concisa el trabajo re...	0.125	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Objetivo 8: Ejemplificar y aplicar estrategias para resolver los ...	0.09	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Figura 32 Tool tip en la tabla Objetivos de CourseEditor v2.0

### 3.2.8 Cambios en la pestaña ‘U.D’

En esta pestaña se definen las unidades didácticas de las que se compone un curso. Cada unidad con su correspondiente *título*, *duración* y *descripción*, se almacena en una tabla de unidades.

Lo cambios realizados en esta pestaña son los que se explican a continuación:

- Se han añadido los botones correspondientes a las opciones ‘*Editar*’ y ‘*Borrar*’ con la misma funcionalidad indicada en la pestaña “Objetivos”
- La duración de la unidad didáctica ahora se puede introducir sin restricción en el número de minutos.

Para este nuevo caso, como el usuario puede introducir tanto números como otros caracteres, se ha agregado una comprobación interna que comprobará si la información introducida en el *jtextField* destinado a la duración de la unidad didáctica, es un número. Si no lo fuera, se mostraría el mensaje de error que se puede observar en la Figura 33.

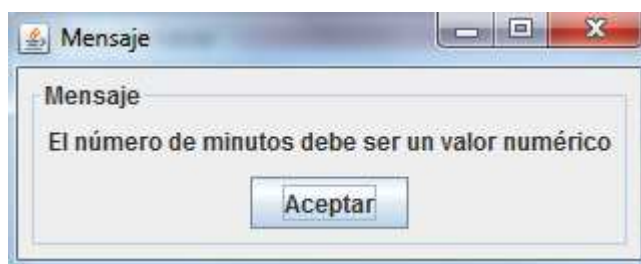


Figura 33 Mensaje de error de CourseEditor v2.0

En las Figuras 34 y 35 se pueden observar los cambios introducidos explicados anteriormente.

**Pantalla Inicial**

Info.General Duración Objetivos **U.D** T.Tareas Tareas Ud-Tareas

**UNIDADES DIDÁCTICAS**

Unidades de las que se compone el curso. Duración entre 1 h 30 min y 2 h.  
Si estima una duración superior divida en epígrafes o unidades más pequeñas.

Título  Duración

Descripción

U.D	Descripción	Duración	Objetivo 1: Interpr...
Introducción y rep...	Presentación del ...	90	<input checked="" type="checkbox"/>
Ejercicios de Pila...	Ejercicios, prese...	90	<input checked="" type="checkbox"/>
Ejercicios de Árb...	Ejercicios	90	<input checked="" type="checkbox"/>

**Ayuda**

Figura 34 Pestaña U.D de CourseEditor v1.0

**Pantalla Inicial**

Archivo Ayuda

Info.General Duración Objetivos **U.D** T.Tareas Roles Tareas Ud-Tareas

**UNIDADES DIDÁCTICAS**

Unidades de las que se compone el curso. Duración entre 1 h 30 min y 2 h.  
Si estima una duración superior divida en epígrafes o unidades más pequeñas.

Título  Duración  min

Descripción

U.D	Descripción	Duración	Editar	Borrar	Objetivo 1:...	Objetivo 2:...	Objetivo 3:...	Objetivo 4:...	Objetivo 5:...	Objetivo 6:...	Objetivo 7:...	Objetivo 8:...
Introducci...	Presentac...	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ejercicios ...	Ejercicios	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ejercicios ...	Ejercicios...	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ejercicios ...	Ejercicios...	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ejercicios ...	Ejercicios	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Presentac...	Presentac...	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Resolució...	Resolució...	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 35 Pestaña U.D de CourseEditor v2.0

## CAPÍTULO: Desarrollo del proyecto

- Por otra parte, en esta tabla también se han introducido descripciones emergentes para visualizar mejor la información como los comentados en la pestaña “Objetivos”.

En las Figuras 36, 37 y 38 se pueden observar ejemplos de la descripción emergente introducida.

U.D	Descripción	Duración	Editar	Borrar	Objetivo 1:...	Objetivo 2:...	Objetivo 3:...	Objetivo 4:...	Objetivo 5:...	Objetivo 6:...	Objetivo 7:...	Objetivo 8:...
Introducci...	Presentac...	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ejercicios ...	Ejercicios	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ejercicios ...	Ejercicios	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ejercicios ...	Ejercicios de Orientación a Objetos e Interfaces				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ejercicios ...	Ejercicios	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Presentac...	Presentac...	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Resolució...	Resolució...	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 36 Tool tip del nombre de la unidad didáctica en la tabla U.D de CourseEditor v2.0

U.D	Descripción	Duración	Editar	Borrar	Objetivo 1:...	Objetivo 2:...	Objetivo 3:...	Objetivo 4:...	Objetivo 5:...	Objetivo 6:...	Objetivo 7:...	Objetivo 8:...
Introducci...	Presentac...	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ejercicios ...	Ejercicios	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ejercicios ...	Ejercicios	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ejercicios ...	Ejercicios, presentación y entrega.				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ejercicios ...	Ejercicios	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Presentac...	Presentac...	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Resolució...	Resolució...	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 37 Tool tip de la descripción de la unidad didáctica en la tabla U.D de CourseEditor v2.0

U.D	Descripción	Duración	Editar	Borrar	Objetivo 1:...	Objetivo 2:...	Objetivo 3:...	Objetivo 4:...	Objetivo 5:...	Objetivo 6:...	Objetivo 7:...	Objetivo 8:...
Introducci...	Presentac...	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ejercicios ...	Ejercicios	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ejercicios ...	Ejercicios	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ejercicios ...	Ejercicios	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ejercicios ...	Ejercicios	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Presentac...	Presentac...	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Resolució...	Resolució...	90			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 38 Tool tip de los objetivos con los que se relaciona cada unidad didáctica en la tabla U.D de CourseEditor v2.0



### 3.2.9 Cambios en la pestaña ‘T.Tareas’

En esta pestaña se definen los tipos de tareas de un curso. Al arrancar la aplicación aparecen una serie de tipos por defecto: Chat, Foro, Consulta, Documentación, Test Multi-respuesta, Ejercicio, Práctica, Votación, Caso de Estudio, Lectura, Compartir Recurso y Proyecto, los cuales, se cargan al arrancar la aplicación desde el constructor de la clase.

También existe la posibilidad de que un usuario cree un nuevo tipo de tarea que se ajuste mejor a sus necesidades pulsando el botón “Definir un nuevo tipo”. Por ejemplo, como parte de una actividad los alumnos tienen que escanear un código QR y acceder a la información que contiene. Para ello, el creador de cursos necesitaría definir una nueva tarea, ‘Escanear código QR’, para introducirlo en la definición de tipos de tarea que contiene el curso. En este caso, se mostrará al usuario una ventana emergente en la cual deberá rellenar la información relativa al nuevo tipo de tarea (*Nombre, Descripción, Iniciales, Servicio síncrono o asíncrono, y seleccionar una imagen* que representará al nuevo tipo de tarea). Además, si el usuario lo desea, también podrá rellenar una plantilla en la que definirá los campos y ayuda que poseerá el nuevo tipo de tarea a la hora de definir tareas concretas.

Los cambios realizados en la pestaña son los siguientes.

- Se han añadido iconos de lápiz y aspa para los botones de “Editar” y “Borrar” de cada tipo de tarea.
- Los marcadores que antes aparecían en la tabla de la pestaña (conceptuales, procedimentales, actitudinales y otros) se asocian normalmente a objetivos, pero no son estáticos para cada tipo de tarea, por lo que se han eliminado de la tabla.
- Se ha ajustado el tamaño de las columnas de la tabla para que se pueda visualizar mejor la información que contiene.

En las Figuras 39 y 40 se pueden observar los cambios introducidos.

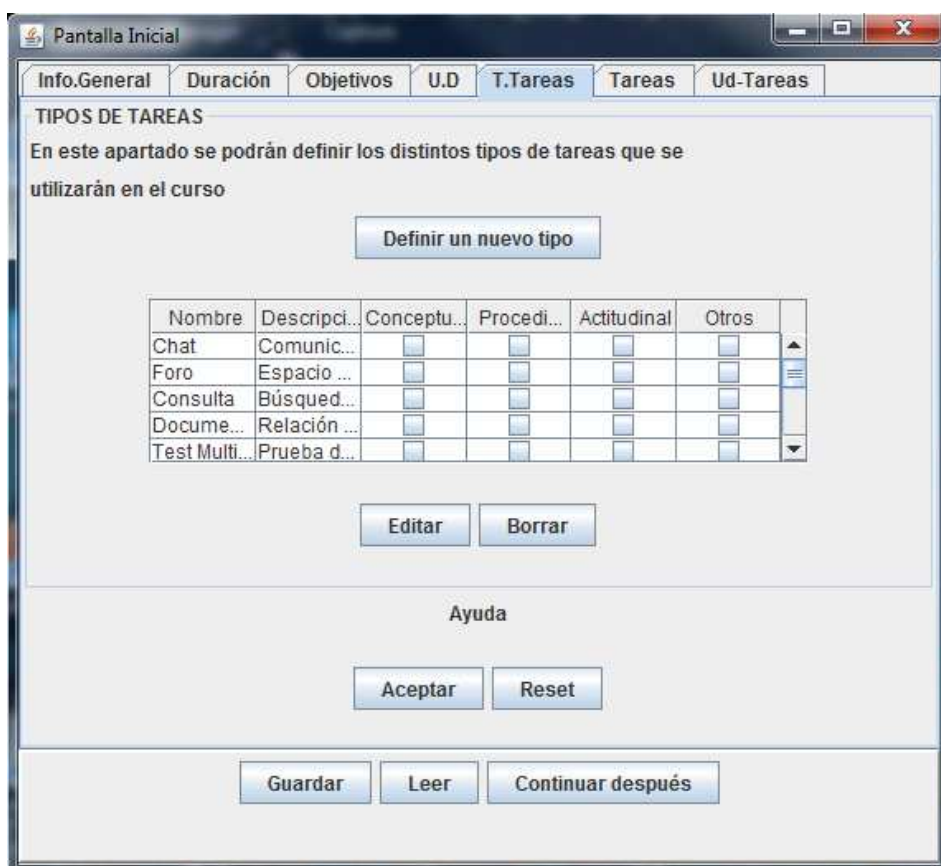


Figura 39 Pestaña T.Tareas de CourseEditor v1.0

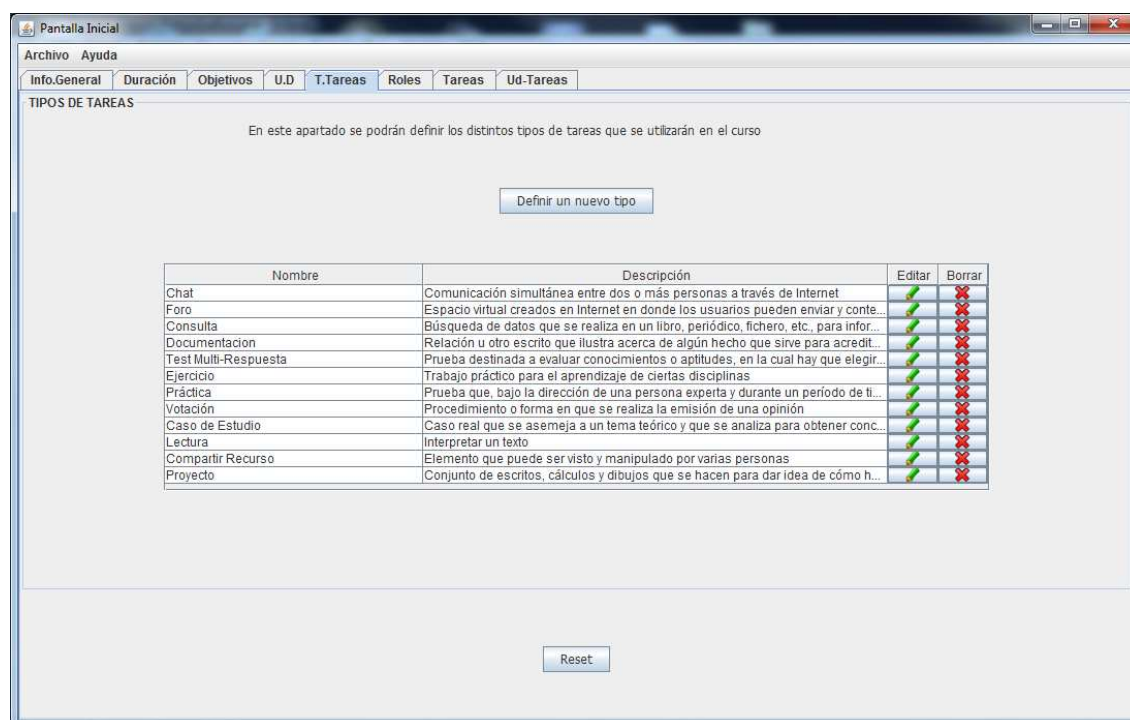


Figura 40 Pestaña T.Tareas de CourseEditor v2.0

- Por otra parte, se han implementado descripciones emergentes del mismo modo que en las pestañas anteriores.

En la Figura 41 se puede observar un ejemplo de descripción emergente.

























Nombre	Descripción	Editar	Borrar
Chat	Comunicación simultánea entre dos o más personas a través de Internet		
Foro	Espacio virtual creados en Internet en donde los usuarios pueden enviar y conte...		
Consulta	Búsqueda de datos que se realiza en un libro, periódico, fichero, etc., para infor...		
Documentación	Relación u otro escrito que ilustra acerca de algún hecho que sirve para acredit...		
Test Multi-Respuesta	Prue... Búsqueda de datos que se realiza en un libro, periódico, fichero, etc., para informarse sobre un asunto		
Ejercicio	Trabajo práctico para el aprendizaje de ciertas disciplinas		
Práctica	Prueba que, bajo la dirección de una persona experta y durante un período de ti...		
Votación	Procedimiento o forma en que se realiza la emisión de una opinión		
Caso de Estudio	Caso real que se asemeja a un tema teórico y que se analiza para obtener conc...		
Lectura	Interpretar un texto		
Compartir Recurso	Elemento que puede ser visto y manipulado por varias personas		
Proyecto	Conjunto de escritos, cálculos y dibujos que se hacen para dar idea de cómo h...		

Figura 41 Implementación de tool tip para una mejor visualización

### 3.2.10 Cambios en la pestaña ‘Tareas’

En la pestaña ‘Tareas’ se definen las tareas que se deben realizar en un curso. Cada tarea tendrá definido un título y una duración. Todas las tareas deben pertenecer a una determinada unidad didáctica y a un tipo de tarea de los descritos anteriormente. Cada tarea definida se almacenará en una tabla de tareas y en un vector de tipo TareaGenerica.

Los cambios que se han introducido en esta pestaña son los siguientes.

- Se han añadido a la tabla los iconos de lápiz y aspa para los botones de “Editar” y “Borrar” de cada tarea.
- Cada vez que se añadía una nueva tarea a la tabla, el usuario debía seleccionar la tarea y pulsar el botón “+ Info”, para introducir la información adicional relativa a cada tarea. De esta forma, el usuario podía olvidar pasar por el botón “+ Info” con todas y cada una de las tareas. Por ello, y para a su vez, minimizar el número de veces que se ha de pulsar un mismo botón, se ha eliminado el mismo y se ha implementado de forma que automáticamente cada vez que se añade una nueva tarea a la tabla, se abre la ventana que permite la introducción de información adicional.
- La duración también se podrá introducir sin ninguna restricción en esta pestaña. Para ello se han cambiado los *jcomboBox* existentes, por campos de introducción de texto (*jTextField*).

**Pantalla Inicial**

Info.General Duración Objetivos U.D T.Tareas **Tareas** Ud-Tareas

**TAREAS**

Es recomendable tener como máximo 4 ó 6 tareas por unidad didáctica

Título: actividad 1 Duración: 5 min

Tipo: Chat U.D: Introducción y repa... **Añadir**

Nombre	Tipo	Duración	U.D	Soporte	Identific...	Objetiv...
activida...	Chat	60	Introdu...	0	17.01.1...	
activida...	Foro	60	Introdu...	0	17.01.1...	
activida...	Consulta	60	Introdu...	0	17.01.1...	

**Borrar** **+ Info.**

**Ayuda**

**Aceptar** **Reset**

**Guardar** **Leer** **Continuar después**

Figura 42 Pestaña Tareas de CourseEditor v1.0

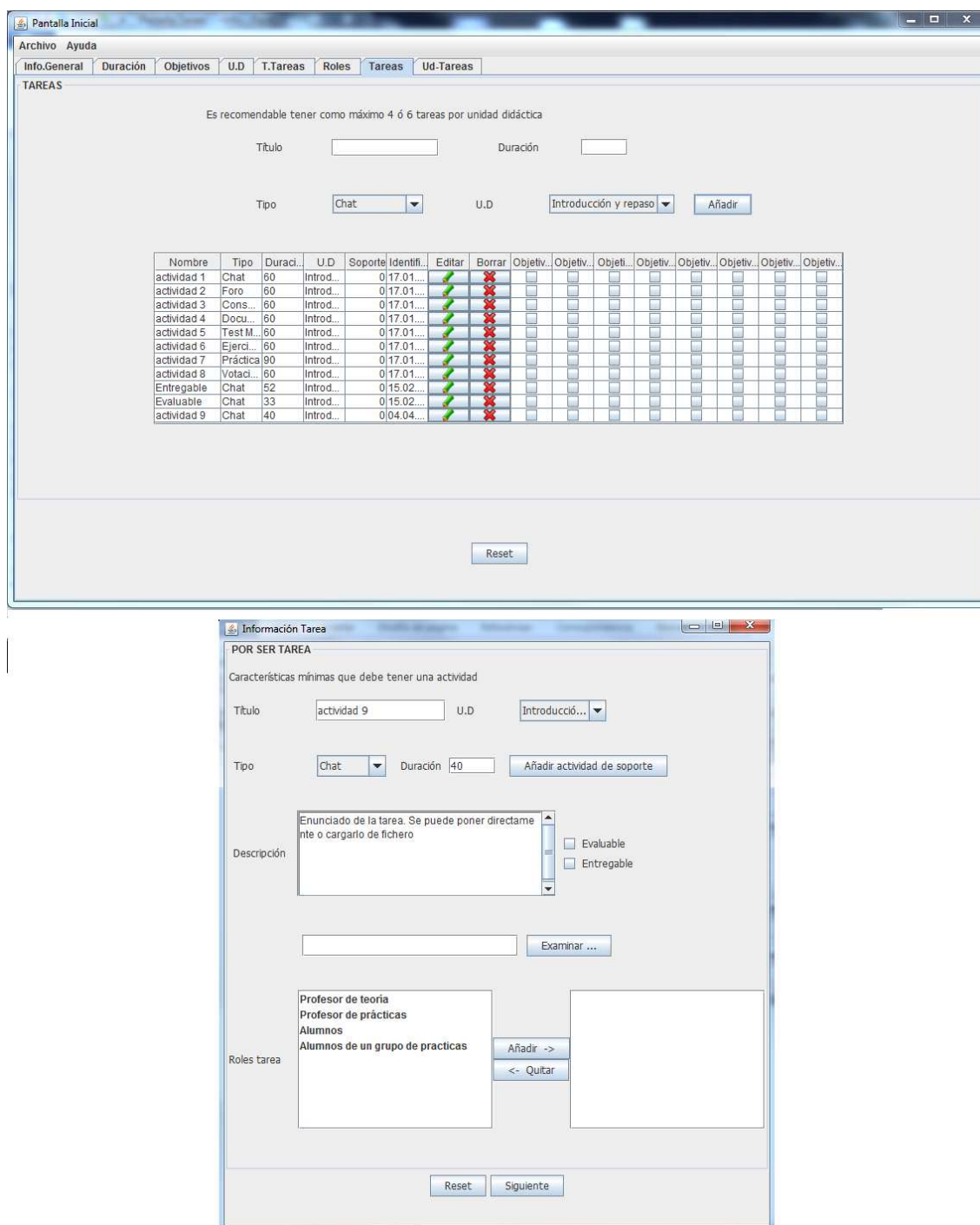


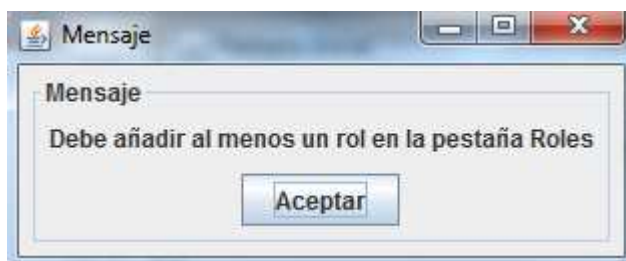
Figura 43 Pestaña Tareas CourseEditor v2.0 y ventana de introducción de información adicional de una tarea que aparecerá automáticamente cuando se pulsa el botón Añadir.

- En el momento de añadir una tarea, se llevarán a cabo una serie de comprobaciones, para que la tarea pueda ser añadida de una forma correcta:
  - En el nuevo campo para añadir la duración del curso de manera no restringida, como el usuario puede introducir tanto números como otros caracteres, se ha agregado una comprobación interna que comprobará si la información introducida en el *jTextField* destinado a la duración de la tarea es correcta. Si no lo fuera, se mostraría un mensaje de error como el que se puede observar en la Figura 44.



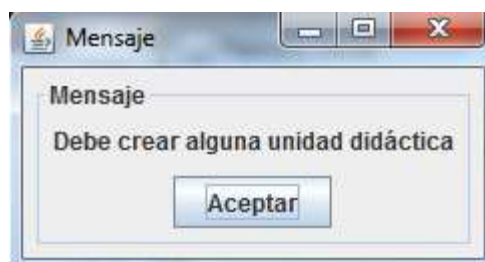
*Figura 44 Mensaje de error. Introducción de un valor no numérico en el campo Duración*

- Como se ha explicado anteriormente, para añadir una tarea, es necesario que haya al menos un rol definido para el curso, por lo tanto, si no se han definido roles en la pestaña anterior, que es la destinada para ello, se mostrará un mensaje de error para recordar al usuario que debe hacerlo.



*Figura 45 Mensaje de error. Debe de haber al menos un rol definido*

- De la misma forma, para la definición de una tarea, es necesario la creación de, al menos, una unidad didáctica a la que la tarea pueda pertenecer, por lo tanto, si el usuario no ha definido una en la pestaña destinada para ello, saldrá un mensaje de error para recordárselo al usuario.



*Figura 46 Mensaje de error. Debe haber definida al menos una unidad didáctica*

- Por otra parte, al igual que en los otros casos en los que el texto es muy largo para mostrarlo completo en la tabla, se han implementado descripciones emergentes que permitan al usuario visualizarlo.

En las Figuras 47, 48, 49, 50 y 51 se pueden observar ejemplos de descripción emergente en esta pestaña.

Nombre	Tipo	Duraci...	U.D	Soporte	Identifi...	Editar	Borrar	Objetiv...	Objetiv...	Objetiv...	Objetiv...	Objetiv...	Objetiv...	Objetiv...	Objetiv...
activid...	Chat	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Chat	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Consu...	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Docu...	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Test M...	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Ejercicio	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Práctica	90	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Votación	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Chat	66	Ejercic...	0	13.03....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Foro	50	Ejercic...	0	13.03....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 47 Tool tip del nombre de la tarea en la tabla Tareas en CourseEditor v2.0

Nombre	Tipo	Duraci...	U.D	Soporte	Identifi...	Editar	Borrar	Objetiv...	Objetiv...	Objetiv...	Objetiv...	Objetiv...	Objetiv...	Objetiv...	Objetiv...
activid...	Chat	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Foro	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Consu...	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Docu...	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Test M...	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Ejerc...	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Práctica	90	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Votación	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Chat	66	Ejercic...	0	13.03....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Foro	50	Ejercic...	0	13.03....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 48 Tool tip del tipo de la tarea en la tabla Tareas en CourseEditor v2.0

Nombre	Tipo	Duraci...	U.D	Soporte	Identifi...	Editar	Borrar	Objetiv...	Objetiv...	Objetiv...	Objetiv...	Objetiv...	Objetiv...	Objetiv...	Objetiv...
activid...	Chat	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Foro	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Consu...	60	Introducción y repaso	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Docu...	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Test M...	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Ejercicio	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Práctica	90	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Votación	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Chat	66	Ejercic...	0	13.03....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Foro	50	Ejercic...	0	13.03....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 49 Tool tip de la unidad didáctica de la tarea en la tabla Tareas en CourseEditor v2.0

Nombre	Tipo	Duraci...	U.D	Soporte	Identifi...	Editar	Borrar	Objetiv...	Objetiv...	Objetiv...	Objetiv...	Objetiv...	Objetiv...	Objetiv...	Objetiv...
activid...	Chat	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Foro	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Consu...	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Docu...	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Test M...	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Ejercicio	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Práctica	90	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Votación	60	Introdu...	0	17.01....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Chat	66	Ejercic...	0	13.03....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Foro	50	Ejercic...	0	13.03....			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 50 Tool tip del identificador de la tarea en la tabla Tareas en CourseEditor v2.0



Nombre	Tipo	Duraci...	U.D	Soporte	Identifi...	Editar	Borrar	Objetiv...	Objetiv...	Objetiv...	Objetiv...	Objetiv...	Objetiv...	Objetiv...	Objetiv...
activid...	Chat	60	Introdu...		0 17.01...			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Foro	60	Introdu...		0 17.01...			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Consu...	60	Introdu...		0 17.01...			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Docu...	60	Introdu...		0 17.01...			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Test M...	60	Introdu...		0 17.01...			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Ejercicio	60	Introdu...		0 17.01...			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Práctica	90	Introdu...		0 17.01...			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Votación	60	Introdu...		0 17.01...			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Chat	66	Ejercic...		0 13.03...			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activid...	Foro	50	Ejercic...		0 13.03...			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 51 Tool tip del objetivo con el que se relaciona la tarea en la tabla Tareas en CourseEditor v2.0

Dentro de la información adicional que el usuario debe de rellenar por cada tarea también se han introducido cambios significativos.

En la versión anterior, las diferentes pantallas que el creador del curso tiene que rellenar por ser tarea y por ser de un tipo además de las que debe rellenar dependiendo de si la tarea es evaluable o entregable, se mostraban mediante pestañas, las cuales se iban haciendo visibles cada vez que el usuario presionaba el botón ‘Aceptar’ para que se guardara la información de la pestaña rellena. Esto podía resultar bastante confuso para el usuario, por lo que en la nueva versión se ha decidido implementar las pantallas de forma consecutiva, de forma que a medida que se rellena cada pantalla se pulsa el botón ‘Siguiente’ (o ‘Finalizar’ si es la última pantalla) y de esta forma se pasa a la siguiente pantalla.

En las Figuras 52, 53, 54 y 55 se muestran cada una de las ventanas de la versión anterior de CourseEditor:

Figura 52 Pantalla ‘Por ser tarea’ de CourseEditor v1.0



**Información Tarea**

Por ser tarea | **Por ser de un tipo** | Por ser evaluable | Por ser entregable

**POR SER DE UN TIPO**

La actividad por pertenecer a un determinado tipo de tarea debe tener unos campos propios que fueron definidos en la pantalla Definir Plantilla

Título

Descripción

Enunciado

Instrucciones

Normas

Objetivos

Ayuda

Aceptar Reset

Guardar

Figura 53. Pantalla 'Por ser de un tipo' de CourseEditor v1.0

**Información Tarea**

Por ser tarea | Por ser de un tipo | **Por ser evaluable** | Por ser entregable

**POR SER EVALUABLE**

Características que debe poseer una actividad si es evaluable

Criterios  Puntuación

Solución

Feedback

Escala: Va de 0 a 100

Ayuda

Aceptar Reset

Guardar

Figura 54 Pantalla 'Por ser evaluable' de CourseEditor v1.0



The screenshot shows a window titled 'Información Tarea' with four tabs: 'Por ser tarea', 'Por ser de un tipo', 'Por ser evaluable', and 'Por ser entregable'. The 'Por ser evaluable' tab is selected, displaying the title 'POR SER EVALUABLE' and the subtitle 'Características que debe poseer una actividad si es entregable'. Below this, there are two date selection fields: 'Fecha Disponible' and 'Fecha Entrega', each with a day dropdown (both set to '1'), a month dropdown (both set to 'Enero'), and a year dropdown (both set to '2006'). There are also two text input fields labeled 'Qué entregar' and 'Cómo entregar'. A checkbox labeled 'Permitir entrega más tarde' is located below these fields. At the bottom of the window, there are three buttons: 'Ayuda', 'Aceptar', and 'Reset' (grouped together), and a 'Guardar' button at the very bottom.

Figura 55 Pantalla 'Por ser entregable' de CourseEditor v1.0

Cabe destacar de estas nuevas ventanas que para la introducción de las fechas necesarias también se han cambiado los *jcomboBox* en los que se podían definir fechas hasta un determinado año. De esta forma se han añadido calendarios haciendo uso de la librería JCalendar descrita anteriormente.

A continuación, en las Figuras 56 y 57 se muestran las pantallas de la nueva versión. En ellas se puede observar la definición de una de las tareas contenidas en el curso de ejemplo descrito en el comienzo del capítulo:

Información Tarea

**POR SER TAREA**

Características mínimas que debe tener una actividad

Título: Examen parcial U.D.: Tema1

Tipo: Test M... Duración: 90 Añadir actividad de soporte

Descripción: Examen parcial para evaluar el conocimiento adquirido por los alumnos al finalizar la primera parte de la asignatura.

☒ Evaluable  
☒ Entregable

Examinar ...

**Roles tarea**

Profesor de prácticas  
Alumnos de un grupo de practicas  
Profesor de teoría

Añadir ->  
<- Quitar

**Alumnos**

Reset Siguiente

Figura 56 Pantalla 'Por ser tarea' de CourseEditor v2.0

Información Tarea

**POR SER DE UN TIPO**  
La actividad por pertenecer a un determinado tipo de tarea debe tener unos campos propios que fueron definidos en la pantalla Definir Plantilla

Título: Examen parcial

Enunciado: adquirido por los alumnos al finalizar la primera parte de la asignatura.

Objetivos: Afianzar los contenidos de la asignatura cursados.

Opciones Avanzadas Reset Siguiete

Figura 57 Pantalla 'Por ser de un tipo' de CourseEditor v2.0

La información adicional que se debe rellenar para cada tarea por ser una tarea comprende una cantidad de campos que no son obligatorios que en la versión anterior el usuario tenía que rellenar obligatoriamente. Estos campos se han cambiado a otra ventana adicional que aparece cuando se pulsa el botón Opciones Avanzadas, de forma que el usuario tiene la libertad de decidir si rellenarlo o no.

Información Tarea - Opciones Avanzadas

**INFORMACIÓN ADICIONAL**  
Características adicionales que debe tener una actividad

Descripción: Descripción de la tarea a realizar en el examen.

Instrucciones: Instrucciones para la realización del examen.

Normas: Normas de realización del examen.

Recurso: Examinar ...

Aceptar Reset

Figura 58 Ventana de información adicional de una tarea de CourseEditor v2.0

Información Tarea

**POR SER EVALUABLE**

Características que debe poseer una actividad si es evaluable

Criterios  Puntuación\*

Nota.....10

Solución

Feedback

\*Escala de la puntuación: Va de 0 a 100

Figura 59 Pantalla 'Por ser evaluable' de CourseEditor v2.0

Información Tarea

**POR SER ENTREGABLE**

Características que debe poseer una actividad si es entregable

Fecha Disponible

Fecha Entrega

Qué entregar

Cómo entregar

☐ Permitir entrega más tarde

Figura 60 Pantalla 'Por ser entregable' de CourseEditor v2.0

### 3.2.11 Cambios en la pestaña ‘Ud-Tareas’

La pestaña Ud-Tareas es la encargada de la secuenciación de las unidades didácticas y la selección del tipo de sincronización que se dará al curso. También sirve de enlace para llegar al Editor Gráfico, en el cual, se secuenciarán las tareas.

Hay tres tipos de sincronización que el usuario puede asignar al curso:

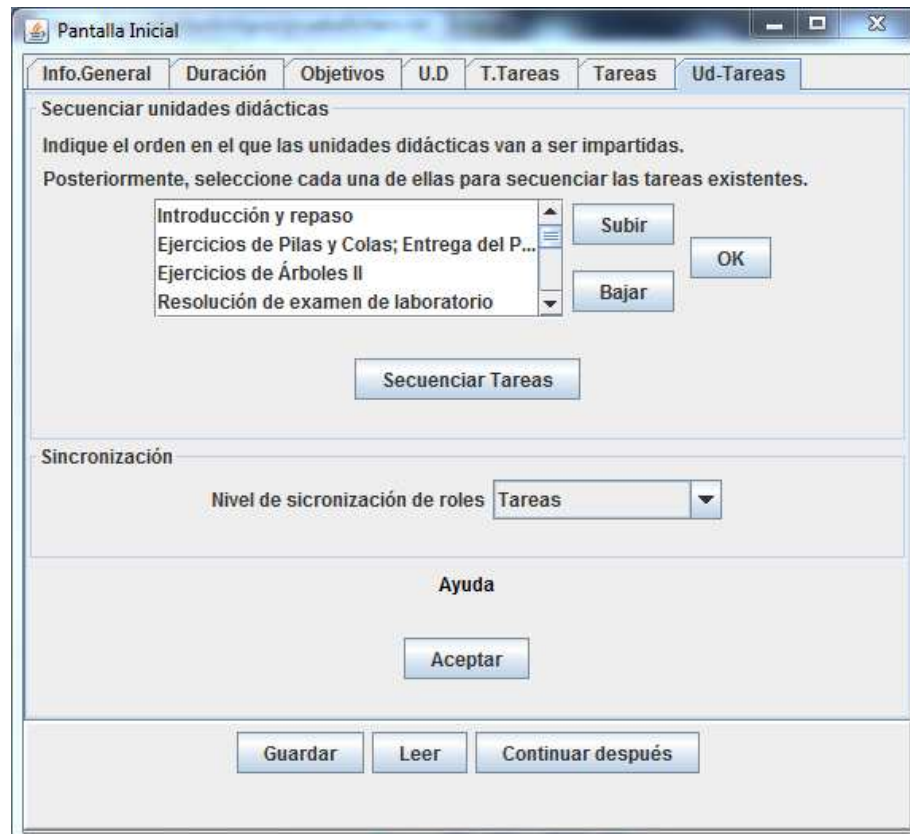
- **Curso Completo:** todos los roles deben encontrarse cuando termine el curso y haber finalizado todas sus tareas.
- **Unidades Didácticas:** cada vez que se termine una unidad, todos los roles deben encontrarse y haber terminado todas sus tareas para poder comenzar con la siguiente.
- **Tareas:** cada vez que se termine un grupo de tareas todos los roles deben encontrarse y haber terminado sus tareas para poder empezar con el siguiente bloque de tareas.

Los cambios que se han realizado en esta pestaña son:

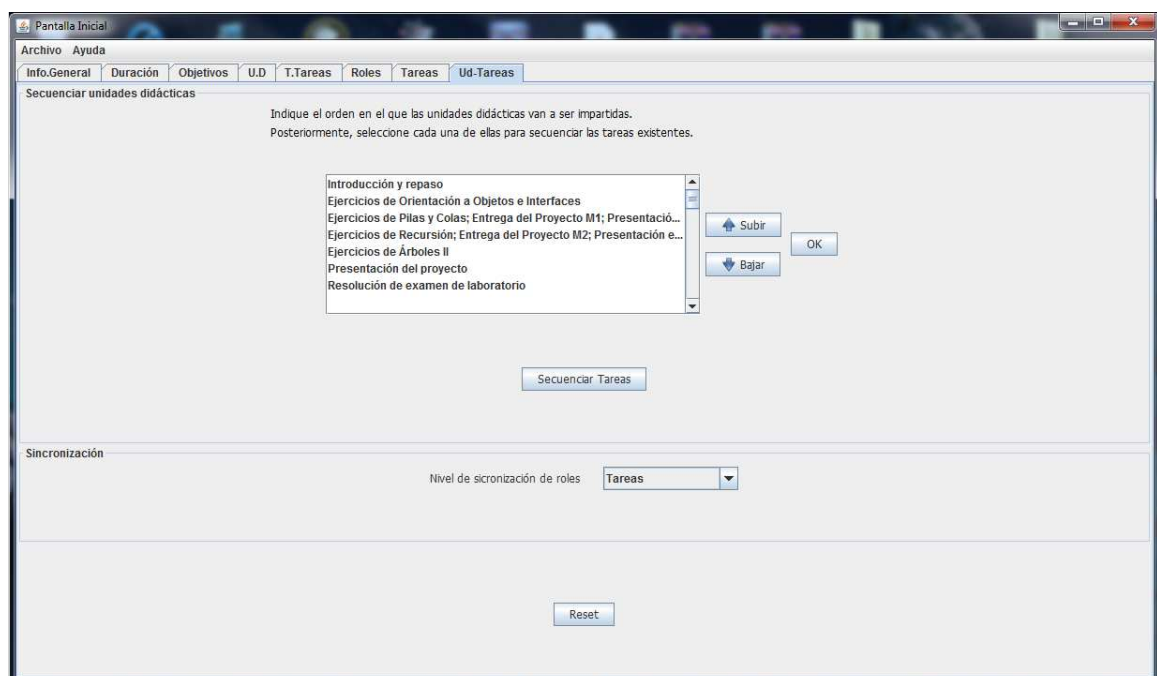
- Se han añadido iconos de flechas hacia arriba y hacia abajo a los botones de ‘Subir’ y ‘Bajar’ de forma que resulte más clara la acción que realizan.
  - Si se pulsa el botón *Subir*, la unidad seleccionada de la lista sube una posición.
  - Si se pulsa el botón *Bajar*, la unidad seleccionada de la lista desciende una posición.

De esta forma, se permite reubicar las unidades didácticas en el orden deseado por el creador de cursos.

- Se ha añadido el botón de Reset que no estaba en la versión anterior.



*Figura 61 Pestaña Ud-Tareas de CourseEditor v1.0*



*Figura 62 Pestaña Ud-Tareas de CourseEditor v2.0*

### 3.2.12 Cambios en el Editor Gráfico

Cuando un usuario ha creado todas las unidades didácticas y tareas de las que se compondrá el curso, tiene la posibilidad de acceder a un editor gráfico, el cual le permitirá secuenciar las tareas que se llevarán a cabo en todas y cada una de las unidades didácticas.

De esta forma, el editor está construido en base a diferentes niveles de granularidad. De esta forma, es necesario que haya definidas actividades o tareas que el alumno tendrá que realizar a lo largo de la duración del curso. Una, o varias de estas tareas compondrán cada una de las unidades didácticas de las que se compone el curso.

El proceso llevado a cabo para la secuenciación es el siguiente: en primer lugar, las unidades didácticas que han sido creadas se ordenan en la pestaña UD-Tareas de la interfaz. En segundo lugar, se han de ordenar las tareas de todas las unidades didácticas que componen el curso usando el editor gráfico. Para ello, se selecciona la unidad didáctica deseada y se pulsa el botón ‘Secuenciar Tareas’.

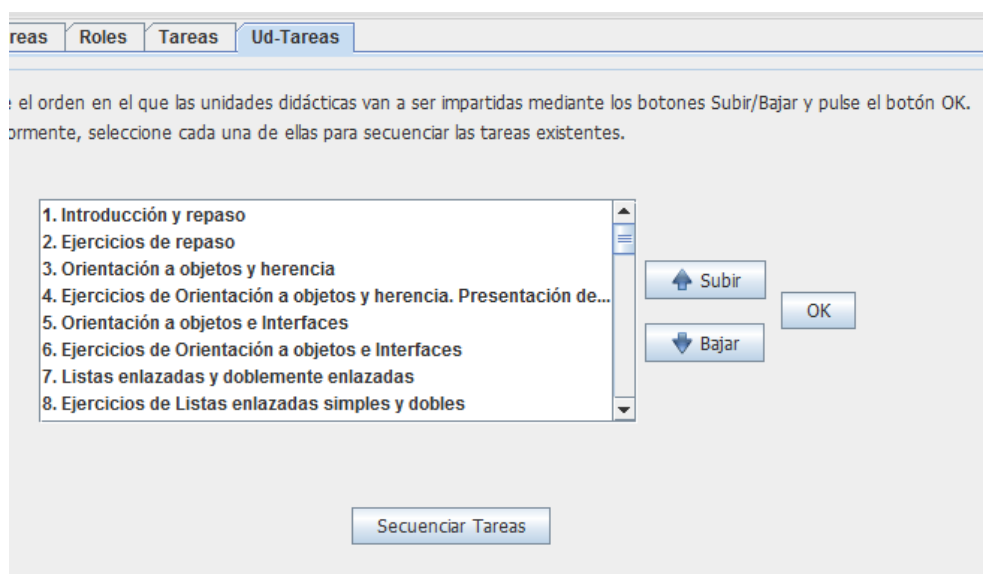


Figura 63. Ordenación de las unidades didácticas de CourseEditor v2.0

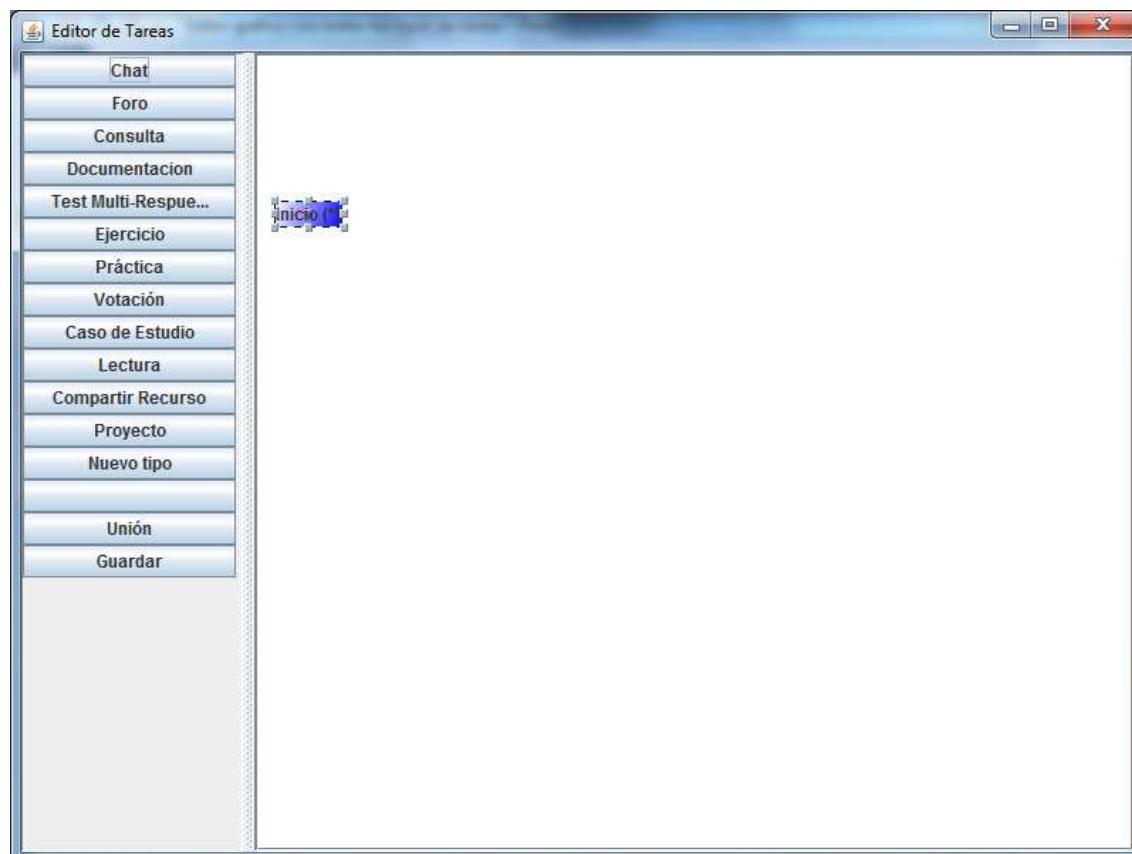
De esta forma, una determinada ejecución del editor representará una unidad didáctica del curso. Los tipos de tareas correspondientes a las tareas de las que se compone la unidad didáctica vendrán representadas por cajas que aparecerán en la pantalla gráfica.

Para realizar la secuenciación, el usuario podrá unir estas cajas con flechas mediante *drag and drop*.

Los cambios efectuados en el editor gráfico son:



Se han eliminado todos los botones que nos permitían obtener las cajas de los distintos tipos de tarea definidos de la parte izquierda del editor. Ahora, todas las tareas definidas para dicha unidad didáctica aparecerán de forma automática en el editor, de forma que el usuario solo tenga que editarlas y moverlas a su gusto para generar la secuenciación gráfica. En las Figuras 64 y 65 se puede observar estas diferencias:



*Figura 64 Editor gráfico de CourseEditor v1.0*

Como podemos observar, en la versión anterior aparecía únicamente la celda de inicio y una serie de botones por cada tipo de tarea, de forma que el usuario tenía que pulsar a cada botón tantas veces como tareas de ese tipo hubiera definidas para esa unidad didáctica en concreto.

La celda de inicio no representa una actividad a realizar, sino que es una caja que aparecerá en todas las secuenciaciones, ya que cada unidad didáctica tiene que comenzar en una celda única de inicio y acabar en una sola tarea final.

La Figura 65 muestra el editor gráfico de una unidad didáctica en la que hay definida una tarea de todos los tipos de tareas por defecto y un nuevo tipo de tarea definido por el usuario.

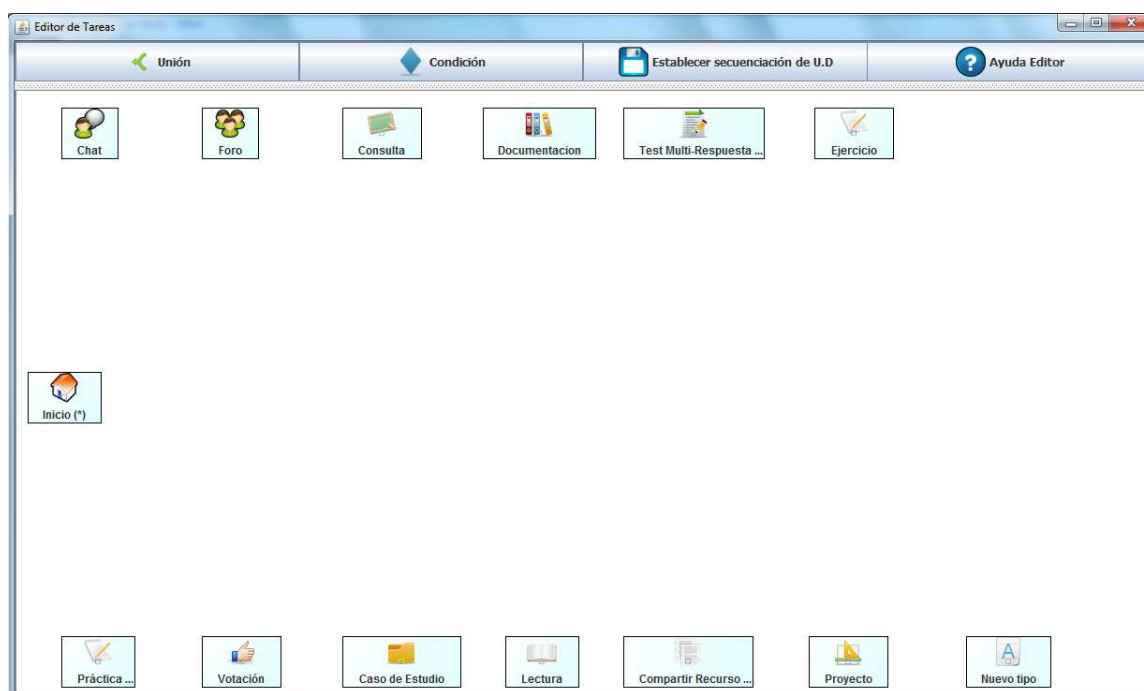
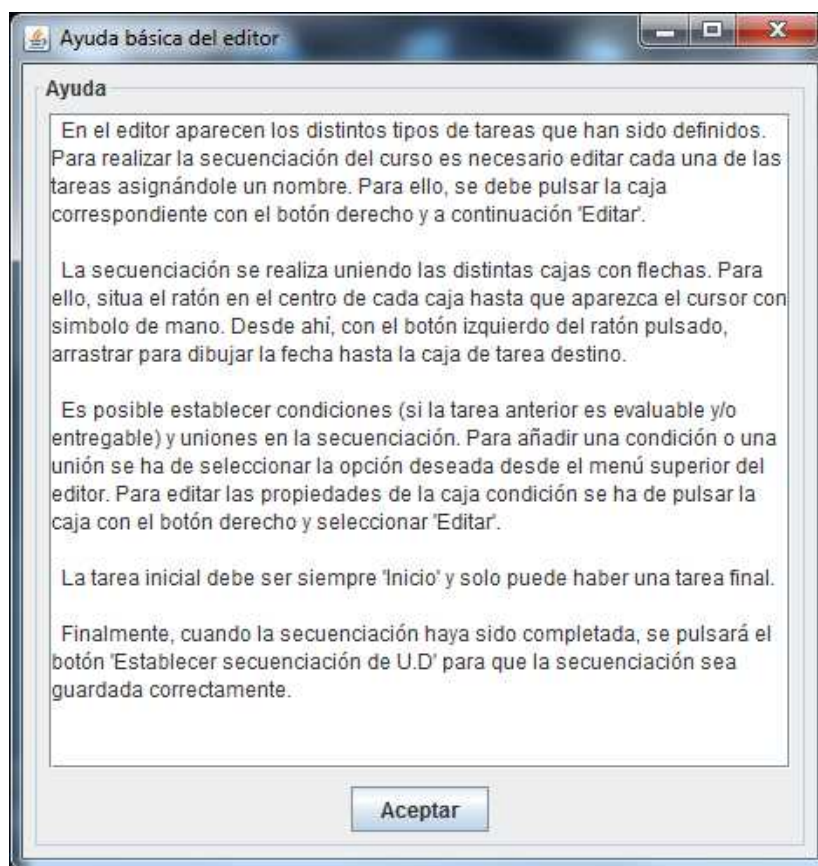


Figura 65 Editor gráfico de CourseEditor v2.0

Tal y como se puede observar en la Figura 65, los botones necesarios para la secuenciación aparecen ahora en la parte superior del editor, y son los que se describen a continuación.

- Boton Unión: nos permite la introducción de una celda de unión en la secuenciación gráfica. De esta forma, el creador de cursos puede establecer que se realicen dos tareas de forma paralela, por ejemplo, podría haber dos grupos de alumnos que, dependiendo de sus preferencias, pueden decidir entre dos tareas a realizar (es decir, es obligatorio que los alumnos realicen una de las tareas, pero no ambas)
- Botón *Condición*: nos permite la introducción de una celda de condición en la secuenciación gráfica.
- Botón *Establecer secuenciación de U.D*: guarda la secuenciación gráfica en un fichero temporal con extensión XML, haciendo uso de la librería Xstream descrita anteriormente.
- Botón *Ayuda Editor*: Muestra una ventana de ayuda básica del editor para los usuarios que no estén previamente familiarizados con la forma de utilizar el editor gráfico. Esto permite al usuario acceder a la ayuda de esta parte de la herramienta de forma rápida y sin necesidad de tener que buscar ayuda acerca del editor gráfico en la ayuda general del menú de la aplicación.

En la siguiente Figura 66 se puede observar la ventana de ayuda mostrada:



*Figura 66 Ayuda básica para un usuario sin experiencia previa en el uso del editor gráfico de CourseEditor v2.0*

A todos estos botones se les ha añadido un icono que permite al usuario un reconocimiento más fácil de los mismos.

En segundo lugar, se ha cambiado la interfaz de las cajas, sustituyendo los diferentes colores que aparecían en la versión anterior, por iconos representativos del tipo de tarea en cuestión. De esta forma se reconoce la función de la caja de forma más sencilla. Los iconos utilizados para cada uno de los tipos son iconos con extensión .png de dimensiones 32x32. [35]

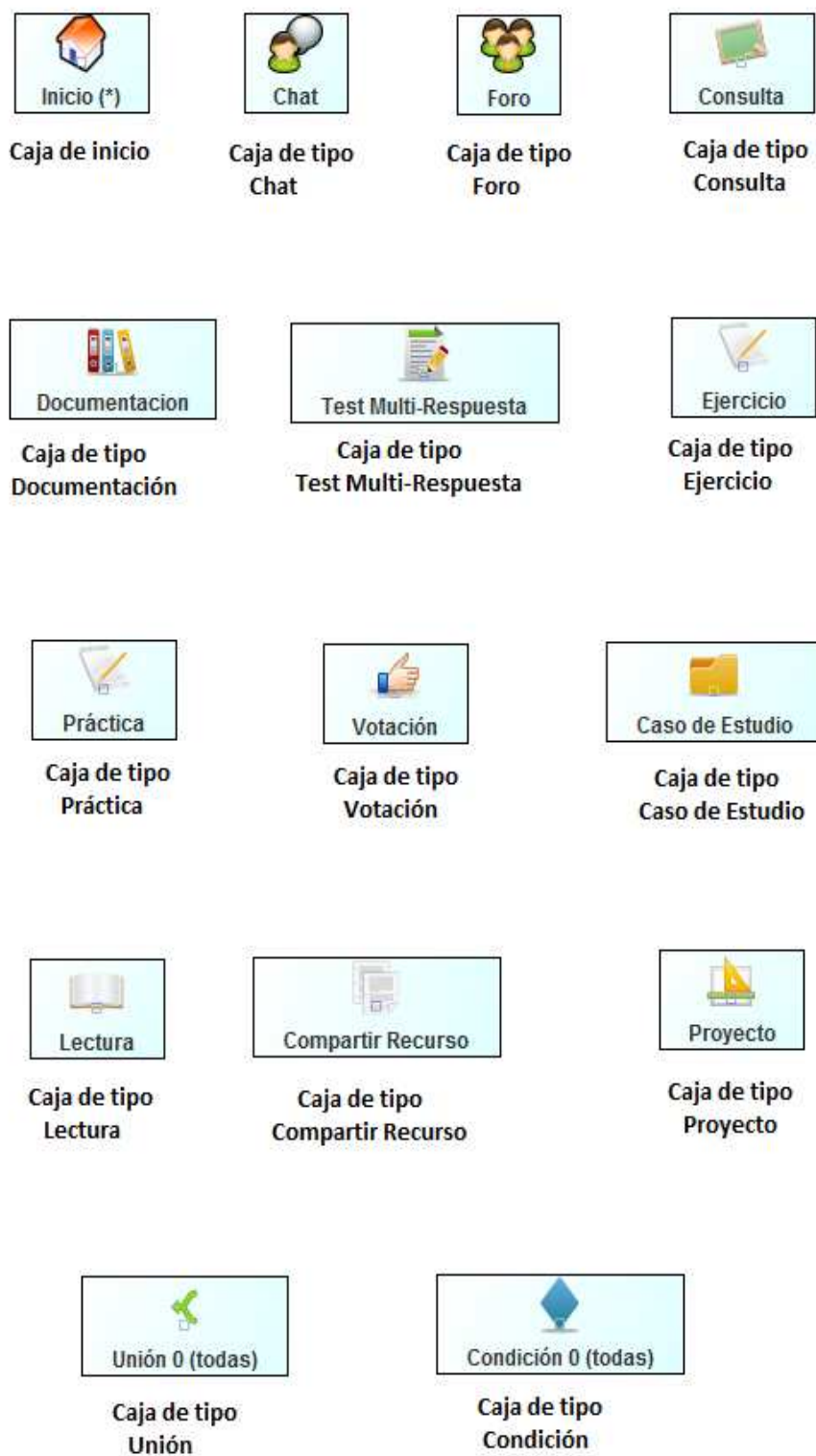
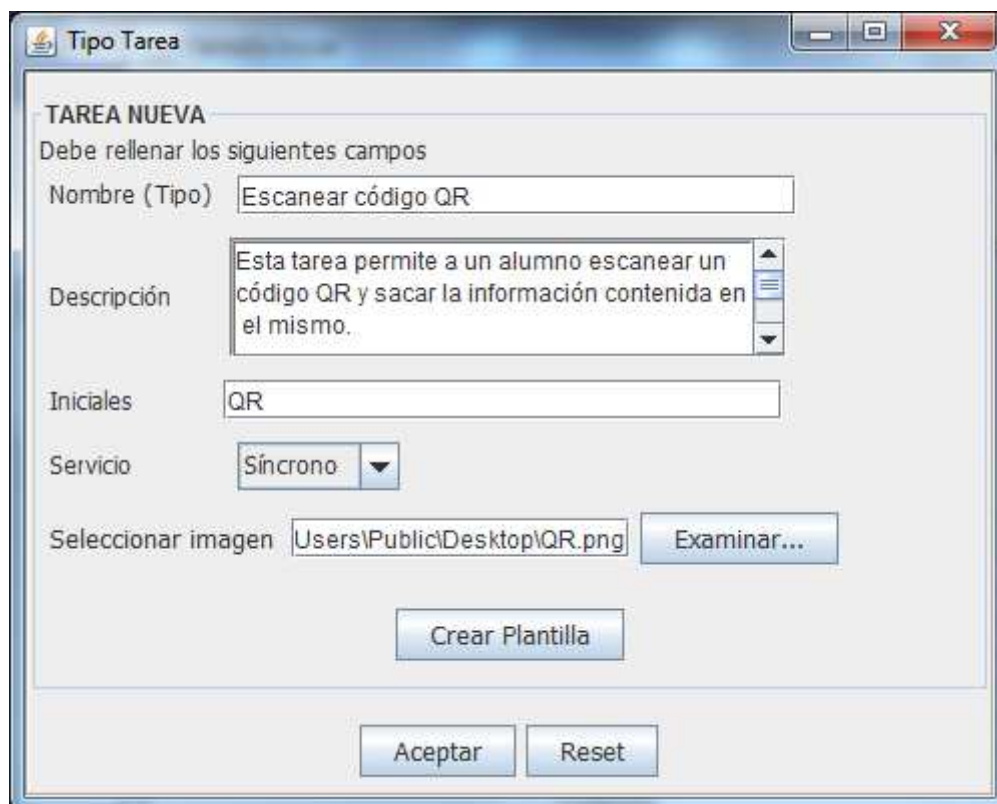


Figura 67 Visualización detallada de los iconos para cada uno de los tipos de tarea.

También el usuario tiene la posibilidad de añadir nuevos tipos de tareas en la pestaña T. Tareas. Los nuevos tipos de tareas también poseerán un icono representativo. Para ello, en la ventana de definición de un nuevo tipo de tarea se ha añadido un campo en el que se permitirá al usuario elegir el icono que desea para cada tipo de tarea definido.

Recuperando el ejemplo de nuevo tipo de tarea ‘Escanear código QR’ mencionado anteriormente, en la Figura 68 se puede visualizar la información que el usuario debe introducir en la ventana.




The screenshot shows a Windows-style window titled 'Tipo Tarea'. Inside, there's a section titled 'TAREA NUEVA' with the instruction 'Debe rellenar los siguientes campos'. The form contains the following fields and controls:

- Nombre (Tipo):** A text box containing 'Escanear código QR'.
- Descripción:** A text box containing 'Esta tarea permite a un alumno escanear un código QR y sacar la información contenida en el mismo.'.
- Iniciales:** A text box containing 'QR'.
- Servicio:** A dropdown menu with 'Síncrono' selected.
- Seleccionar imagen:** A text box containing the file path 'Users\Public\Desktop\QR.png' and an 'Examinar...' button next to it.
- Buttons:** 'Crear Plantilla' (centered), 'Aceptar' (bottom left), and 'Reset' (bottom right).

*Figura 68 Ventana de definición de un nuevo tipo de tarea*

La imagen seleccionada por el usuario debe ser un icono de dimensiones 32x32 o inferiores. Si la imagen seleccionada no cumple estas restricciones, al pulsar el botón ‘Crear plantilla’ (o directamente el botón ‘Aceptar’ si no se desea crear una plantilla para el nuevo tipo) nos aparecerá un mensaje de error informativo.



The screenshot shows a small error dialog box titled 'Mensaje'. It contains the text: 'La imagen seleccionada debe ser de dimensión 32x32 o inferior'. There is an 'Aceptar' button at the bottom.

*Figura 69 Mensaje de error tras una introducción errónea de imagen para el tipo nuevo*

No obstante, si no se elige un icono para la tarea, ésta aparecerá con un icono por defecto para nuevas tareas. El icono por defecto es el que se puede observar en la Figura 70.



*Figura 70 Icono por defecto para los nuevos tipos de tarea definidos por el usuario en CourseEditor v2.0*

Como se puede observar en el ejemplo mostrado en las Figuras 71 y 72, para completar la unidad didáctica el alumno debe leer artículo que introduce los temas a tratar, así como realizar una documentación y una consulta del mismo. Después, el alumno tendrá la posibilidad de participar en una de las tres siguientes tareas (participación en un foro de la asignatura, en un chat, o compartir un recurso). Una vez realizada una de las tres tareas anteriores, el alumno deberá escanear un código QR del cual extraerá una información que le permitirá, o bien realizar una votación o bien un caso de estudio de la misma. Posteriormente el alumno realizará un ejercicio que permitirá aplicar y reforzar los contenidos aprendidos. Más tarde, realizará un test o examen en el cual se medirá de forma objetiva lo que ha aprendido y finalmente realizará una práctica y un proyecto final.

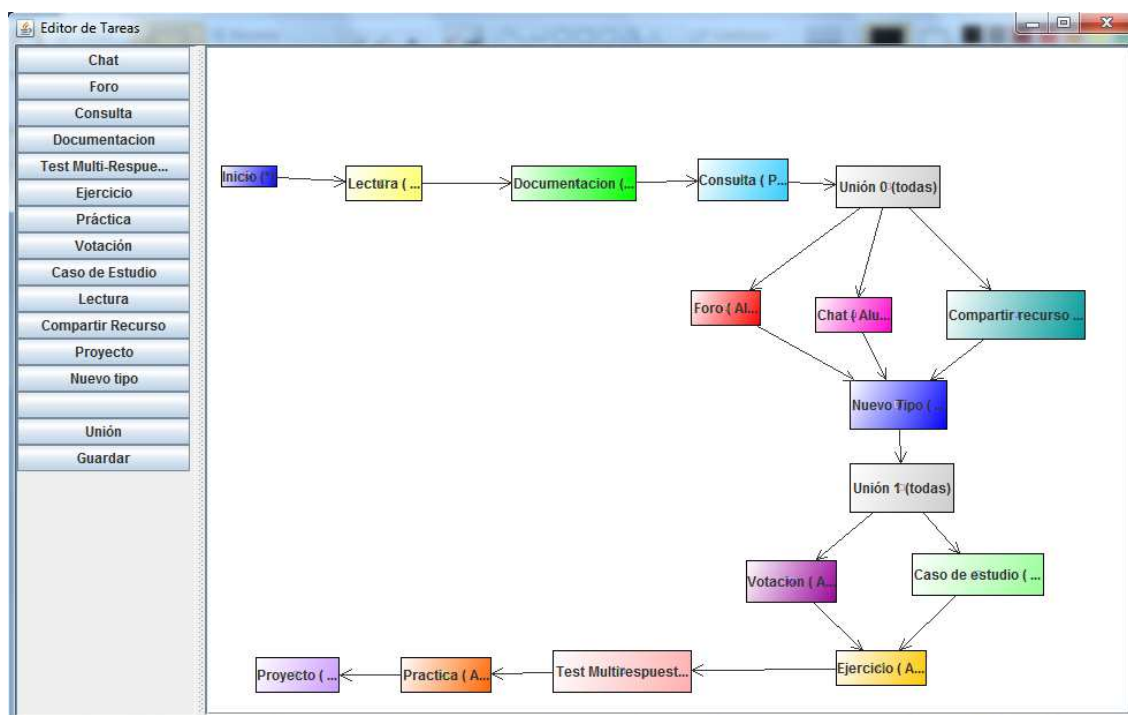


Figura 71 Representación gráfica de CourseEditor v1.0

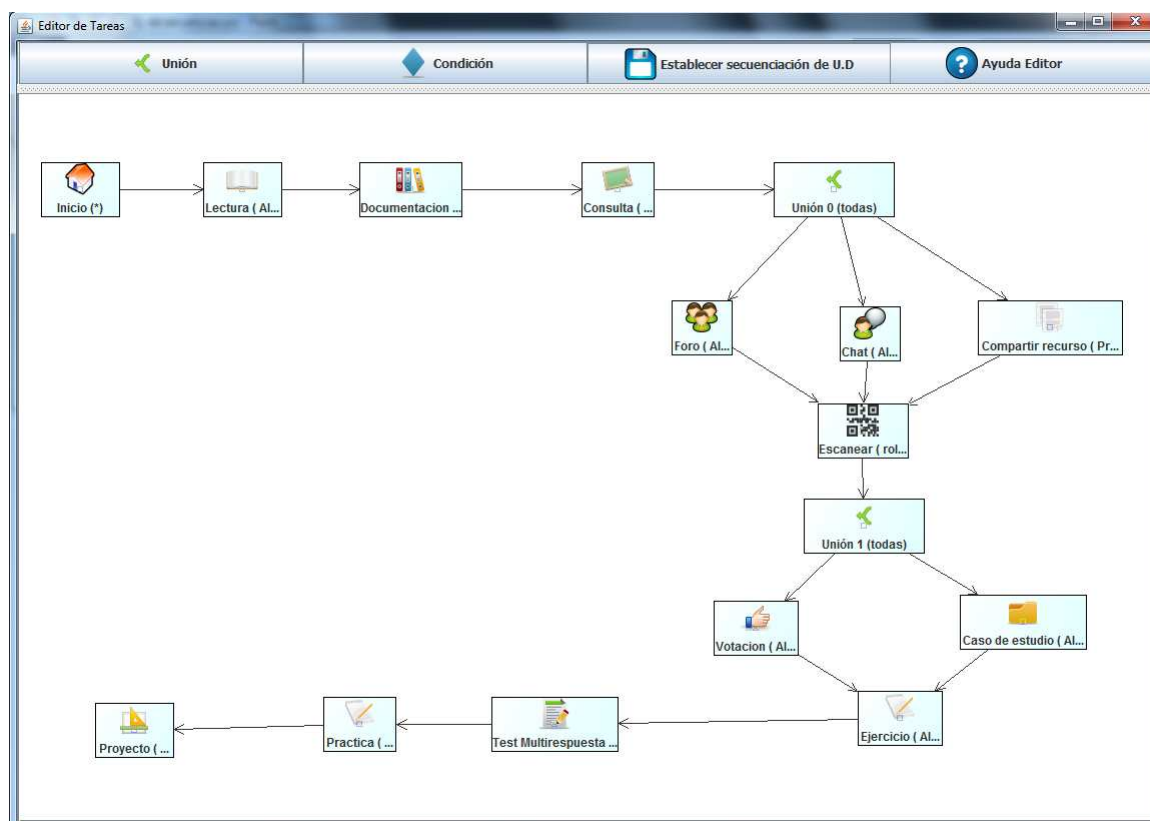


Figura 72 Representación gráfica de CourseEditor v2.0

## 3.3 Errores en la herramienta

### 3.3.1 Problemas de aparición de información en los *jcomboBox*

En la pestaña ‘Tareas’ de la versión anterior de la herramienta, al intentar añadir una nueva tarea sin haber presionado el botón “Aceptar” de cada una de las pestañas anteriores con el fin de guardar la información correspondiente a las unidades didácticas y los tipos de tareas, estas no se cargaban en los *jcomboBox*.

Esto se ha solucionado, tal y como se ha mencionado en el apartado anterior, gracias a la implementación de forma que cada vez que se cambia de pestaña se guarda automáticamente la información introducida sin necesidad de tener que pulsar un botón cada vez que se cambia de pestaña, ya que, además de ser bastante engorroso para el creador de cursos, puede dar lugar a olvidos.

### 3.3.2 Desaparición de roles al definir nuevas tareas

En la versión anterior en la pestaña ‘Tareas’ al añadir más información a una tarea, presionando el botón “+ Info” con la tarea seleccionada, y añadir un rol, este rol desaparecía y ya no se encontraba disponible para esa tarea ni para el resto. Esto sucedía si primero se definían todas las tareas, y luego se iban editando todas. Para solucionar este problema, se ha procedido a hacer dos cambios.

- Se ha eliminado el botón “+Info” y se ha implementado de forma que cada vez que se añada una nueva tarea, salte automáticamente la ventana de definición de información adicional. Esto se ha realizado así porque se ha considerado que tiene las siguientes ventajas:
  - Disminución del número de clicks en la herramienta, lo cual mejora considerablemente la usabilidad del sistema.
  - Al tener que pasar todas y cada una de las tareas por el mencionado botón, existía un alto porcentaje de posibilidades de que el usuario olvidara editarlas todas. De esta forma, el usuario la edita en cuanto la define, y no hay lugar a que se olvide una.
  - Al editar cada tarea a la vez que se define, se soluciona el error antes mencionado de la desaparición de roles, de forma que ahora tendremos en todo momento los roles definidos disponibles para todas y cada una de las tareas.
- Se ha habilitado una nueva pestaña llamada ‘Roles’ en la que se definirán todos los roles que van a participar en el desarrollo del curso definido. Esta ventana se explica más en detalle en el siguiente apartado.

### 3.3.3 Nueva pestaña: ‘Roles’

En la implementación de la versión anterior la definición de roles se realizaba en la ventana para añadir información adicional a las tareas, pulsando el botón “+Info”. Esto



no resultaba muy útil, debido a que si el usuario quería añadir un nuevo rol al curso, debía definir una tarea para poder hacerlo.

En esta nueva pestaña se podrán definir todos los roles requeridos, y estos luego serán cargados en la pestaña de información adicional de las tareas para su posterior utilización.

Debe rellenar los siguientes campos para definir un nuevo rol

Nombre (título asignado al rol)

Descripción

Pertenencia

Etiquetar como (una etiqueta es una caracterización del rol, por ejemplo, si es un grupo de prácticas, de delegados, etc.)

Nombre	Descripción	Pertenencia	Etiqueta	Editar	Borrar
Alumnos	Todos los alumnos matriculados e...	learner	Sin etiqueta		

Figura 73 Nueva pestaña para la definición de Roles de CourseEditor v2.0

Dentro de la pestaña Roles, se tendrá que rellenar la siguiente información:

- Nombre: Nombre con el que se conocerá al rol, por ejemplo, *Profesor de teoría*.
  - Descripción: Un breve resumen del rol, por ejemplo, *Profesor que se encargará de la docencia de los grupos magistrales*.
- Pertenencia: La especificación IMS-LD define roles de dos tipos principales:
  - Learner: Si el rol es un alumno o persona que cursa el curso en cuestión.
  - Staff: Si el rol pertenece a una persona que se encarga de la impartición de la docencia del curso.

En el caso del ejemplo contemplado, la pertenencia de este rol sería *Staff*.

- Etiquetar como: Dentro de cada uno de los tipos de roles, se podrá etiquetarlos de forma que quede definido si el rol en cuestión será una sola persona o un grupo, o si pertenecerán a grupos de prácticas o teoría, etc.

Debe rellenar los siguientes campos

Nombre

Descripción

Pertenencia

Etiquetar como

Nombre	Pertenencia	Etiqueta	Editar	Borrar

Figura 74 Campos que se han de rellenar para definir un nuevo rol de CourseEditor v2.0

Dentro del *jcomboBox* desplegable, se podrá seleccionar alguna de las opciones existentes por defecto o crear una etiqueta nueva, según las preferencias de la persona que cree el curso. Esta nueva etiqueta será recordada por la herramienta durante su ejecución por si se quiere volver a utilizar para otro rol.

Si por algún motivo, el usuario se equivoca en la introducción del nombre de una nueva etiqueta o desea modificarla, se ha habilitado un botón llamado “Borrar” etiqueta que borrará la etiqueta seleccionada en el *jComboBox*.

Cuando pulsamos crear una nueva etiqueta, se abrirá una pequeña ventana en la que se permite introducir el nombre de la nueva etiqueta:

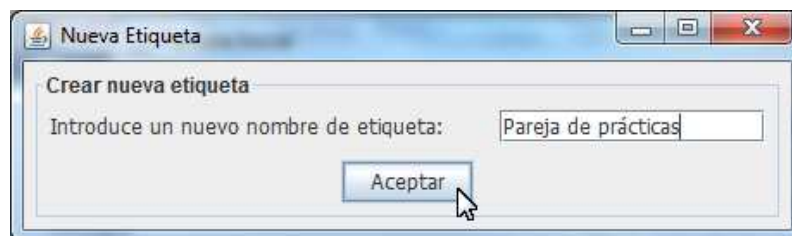


Figura 75 Ventana para la creación de nuevas etiquetas de rol de CourseEditor v2.0

Por otra parte, se han introducido descripciones emergentes como las descritas en las pestañas anteriores para la visualización completa de los campos más extensos.

En la Figura 76 se puede observar un ejemplo de descripción emergente.

Nombre	Descripción	Pertenencia	Etiqueta	Editar	Borrar
Alumnos	Todos los alumnos matriculados e...	learner	Sin etiqueta		
Alumnos de un grupo de practicas	Alumnos que forman un grupo de p...	learner	Grupo de prácticas		
Profesor de teoría	Profesores que se encargan de la te...	teacher	Sin etiqueta		
Profesor de practicas	Profesores que se encargan de la p...	teacher	Sin etiqueta		

*Figura 76 Tool tip de la descripción del rol de CourseEditor v2.0*

### 3.3.4 Eliminación de la necesidad de definir tareas soporte en un curso

Cuando se guardaba un curso creado en la herramienta, si el curso estaba incompleto, la herramienta solo generaba un documento .txt, que permitía volver a cargar toda la información que fue rellenada sin necesidad de volverla a introducir en una ejecución posterior. Por otra parte, si la aplicación detectaba que el curso estaba completo, además de generar el documento de texto mencionado, también generaba los ficheros XML (el que sigue la especificación IMS-LD y los tres ficheros de información adicional introducida).

No obstante, se producía un error en la herramienta, de forma que era necesario tener al menos definida una actividad de soporte para el curso para que la herramienta considerara que el curso estaba completo y generara toda la documentación.

Sin embargo, y ya que es perfectamente posible que un curso no posea ninguna actividad de soporte, y aún así el curso debería considerarse como completo, se ha subsanado este error, de forma que ahora se crean los XML tanto si se definen actividades soporte como si no.

Finalmente, cabe destacar que se ha procedido a analizar todas y cada una de las pestañas y ventanas de la herramienta, así como en la ayuda y se han corregido todos los errores ortográficos que poseía.

## 3.4 Nuevas funcionalidades

### 3.4.1 Introducción de funcionalidad de nivel B

Como se ha mencionado en el estado del arte del presente proyecto de fin de carrera, la especificación IMS-LD está dividida en niveles de complejidad creciente (niveles A, B y C). La versión anterior de la herramienta solo cubría las definiciones del nivel A. Sin embargo, para hacer más completa la definición de cursos, se ha decidido ampliar las funcionalidades de la herramienta y se han añadido las dos aportaciones fundamentales

de funcionalidad recogidas en el nivel B de la especificación: condiciones y propiedades. Se ha elegido añadir esta nueva funcionalidad debido a que permite al usuario diseñar ambientes de aprendizaje personalizados así como modelos colaborativos de aprendizaje.

De esta forma, en la secuenciación de las unidades didácticas, un profesor podrá decidir que camino podrá seguir un alumno en función de factores como por ejemplo, los resultados obtenidos en un examen ó si ha entregado o no a tiempo un determinado trabajo.

Para ello, y teniendo en cuenta que todas las tareas pueden ser definidas como entregables, evaluables o ambas, se tendrán en cuentas las siguientes opciones:

- Si es evaluable:
  - Nota mayor que
  - Nota menor que
- Si es entregable:
  - Entregada a tiempo
  - Entregada tarde
  - No entregada

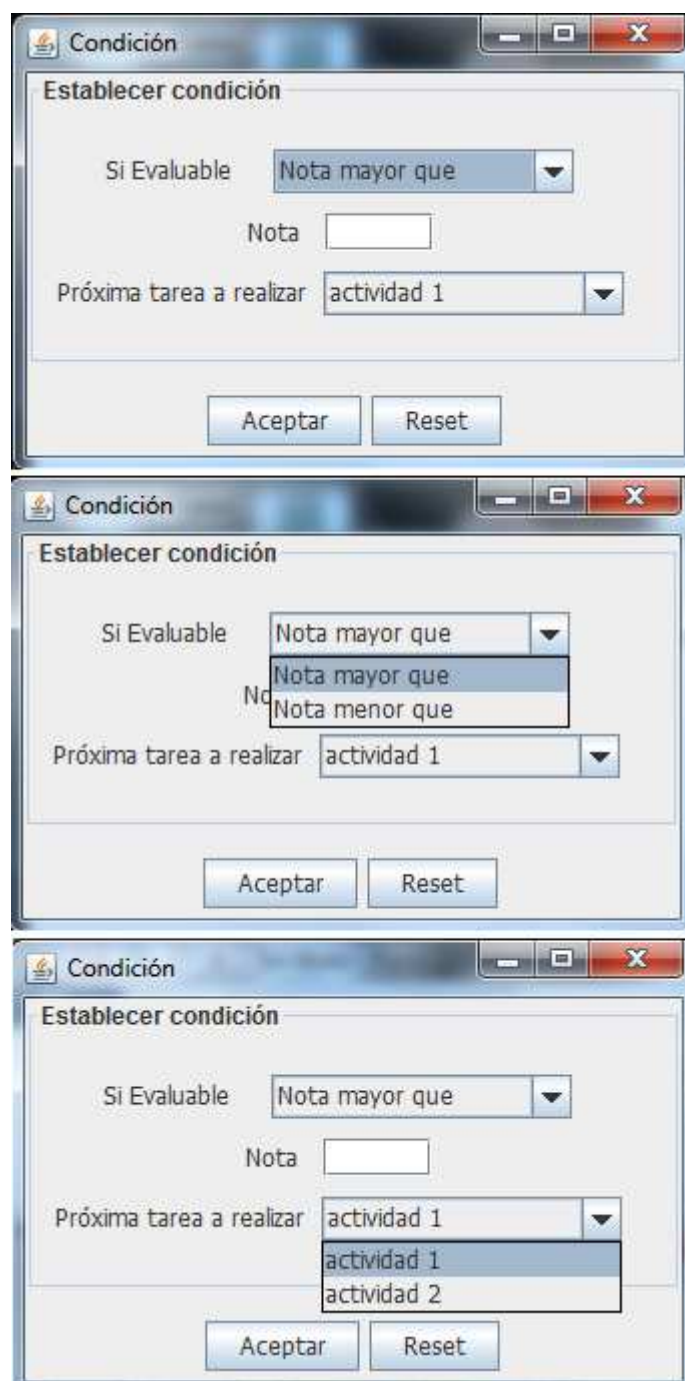
La ventana que se mostrará si la tarea es evaluable y entregable se muestra en la Figura 77.

The figure consists of three vertically stacked screenshots of a dialog box titled 'Condición'. Each screenshot shows the 'Establecer condición' section with the following fields: 'Si Evaluable y Entregable' (a dropdown menu), 'Nota' (a text input field), and 'Próxima tarea a realizar' (a dropdown menu). At the bottom of each dialog are 'Aceptar' and 'Reset' buttons.

- Top screenshot:** The 'Si Evaluable y Entregable' dropdown is set to 'Entrega a tiempo'. The 'Nota' field is empty. The 'Próxima tarea a realizar' dropdown is set to 'actividad 1'.
- Middle screenshot:** The 'Si Evaluable y Entregable' dropdown menu is open, showing a list of options: 'Entrega a tiempo', 'Entrega tarde', 'No entregada', 'Nota mayor que', and 'Nota menor que'. The 'Nota' field is empty. The 'Próxima tarea a realizar' dropdown is partially visible, showing 'ac'.
- Bottom screenshot:** The 'Si Evaluable y Entregable' dropdown is set to 'Entrega a tiempo'. The 'Nota' field is empty. The 'Próxima tarea a realizar' dropdown menu is open, showing a list of options: 'actividad 1' and 'actividad 2'.

Figura 77 Definición de la condición si la tarea es evaluable y entregable de CourseEditor v2.0

La ventana que se mostrará si la tarea es evaluable se muestra en la Figura 78.



*Figura 78 Definición de la condición si la tarea es evaluable de CourseEditor v2.0*

La ventana que se mostrará si la tarea es entregable se muestra en la Figura 79.

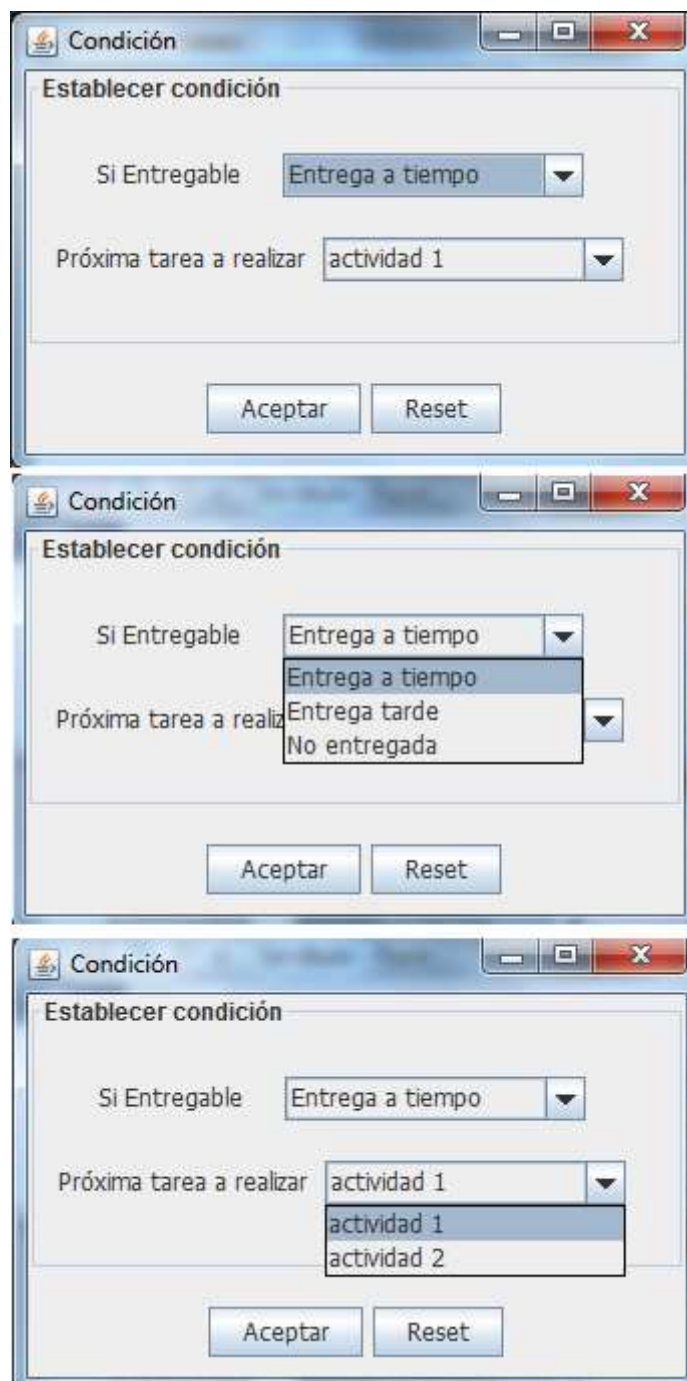


Figura 79 Definición de la condición si la tarea es entregable de CourseEditor v2.0

### **3.4.2 Persistencia de la secuenciación gráfica en el editor**

En la implementación de la versión anterior de la herramienta, al guardar un curso, se guardaba toda la información introducida en la creación del mismo, excepto la secuenciación gráfica de las unidades didácticas.

Se ha implementado una nueva funcionalidad que permite guardar las secuenciaciones gráficas creadas en el editor para cada una de las unidades didácticas, de forma que al pulsar el botón ‘Establecer secuenciación de U.D’ del editor gráfico, se guardará la información de las cajas y flechas en vectores que se utilizarán para crear un fichero XML con dicha información, que permitirá posteriormente la reconstrucción de la secuenciación.



# Capítulo 4

## Implementación del proyecto

En este capítulo se presenta una explicación detallada del diseño e implementación llevados a cabo en la herramienta para la realización de todos los cambios y mejoras introducidos. Posteriormente, se explicarán en detalle las clases que han sufrido modificaciones así como las nuevas clases añadidas.

### 4.1 Diseño e implementación

A continuación se muestra el diseño de la aplicación con el objetivo de que el lector entienda mejor el funcionamiento de la misma.

En la Figura 80 se muestra el diagrama UML de la aplicación. Las clases existentes en la versión 1.0 que han sido modificadas para la implementación de las mejoras se muestran coloreadas de color azul. Las nuevas clases añadidas se muestran en color rojo. En el diagrama no se muestran los atributos ni métodos para facilitar la comprensión de mismo.

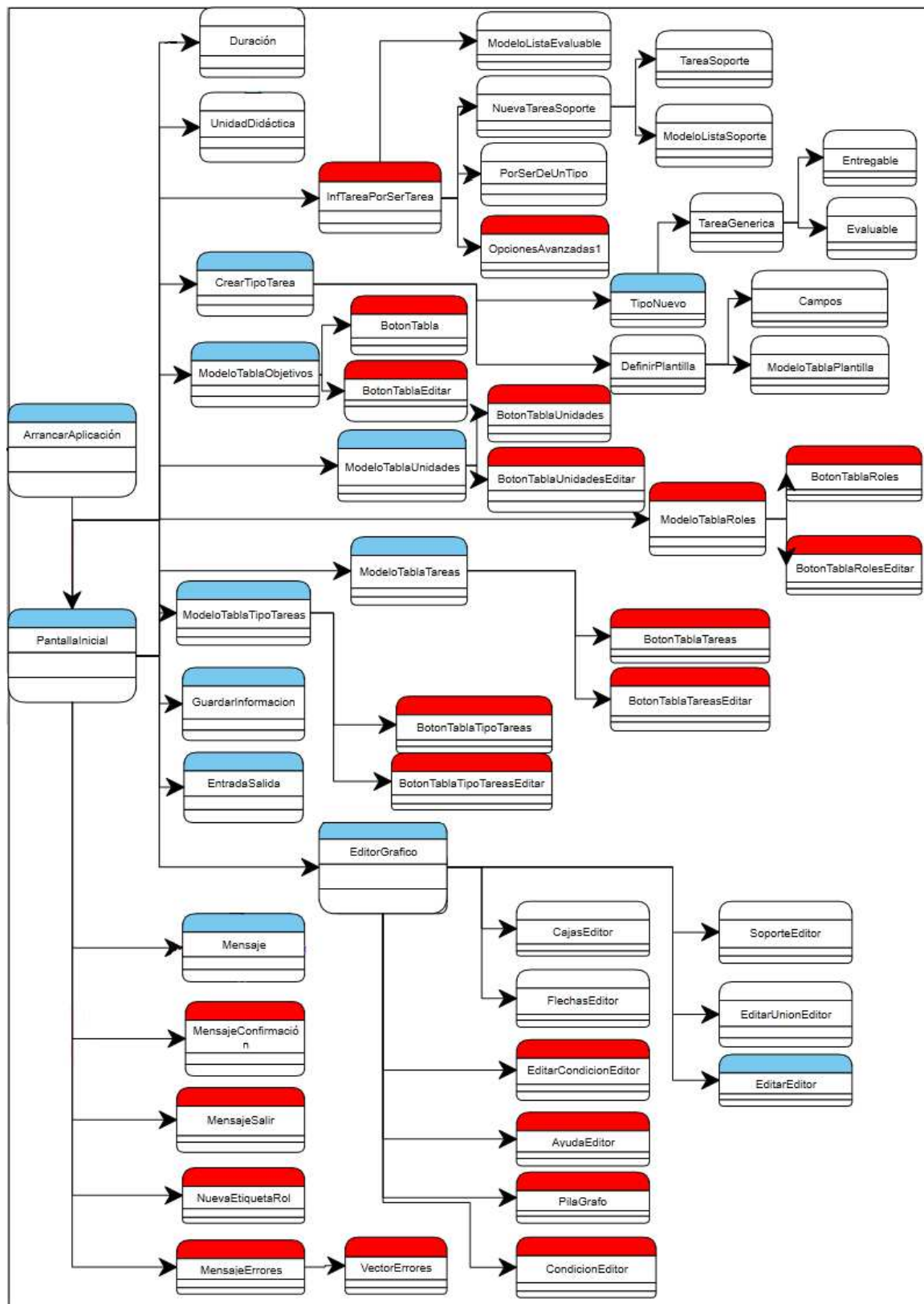


Figura 80 Diagrama UML de la aplicación

En los siguientes apartados se explicará la implementación de cada una de las clases más importantes que han sido modificadas y cada una de las clases que han sido añadidas a la aplicación.

### 4.1.1 Clase PantallaInicial

La clase *PantallaInicial*, la cual inicia la aplicación, es la que define el interfaz de usuario principal de la herramienta. En esta clase se han llevado a cabo importantes modificaciones.

- **Introducción de un menú.**

Tal y como se ha explicado en el apartado de Desarrollo del proyecto se ha introducido un menú en la parte superior izquierda de la herramienta que sustituye la funcionalidad que tenían los distintos botones situados en las distintas pestañas de la herramienta.

Para ello, se ha definido un objeto *jMenu* al cual se le han añadido los diferentes *jMenuItems* que lo componen.

Para añadir la funcionalidad a cada se ha añadido a cada uno de ellos un *actionListener*, el cual, escucha si se está pulsando una opción o otra y actúa en consecuencia.

```
jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
    @Override
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        actionPerformedNuevo(evt);
    }
});
```

Dentro de las opciones disponibles dentro del menú también se han llevado a cabo modificaciones importantes.

En primer lugar, las opciones ‘Guardar’ y ‘Guardar como’ se han implementado de manera que el usuario puede decidir la ubicación donde se guarde la documentación generada por la herramienta.

Para ello se ha definido un objeto *jFileChooser* el cual proporciona una interfaz de usuario que permite al creador de cursos navegar por el sistema de archivos y seleccionar la ubicación deseada para guardar la documentación.

De la misma manera para la opción de ‘Abrir’ se ha definido otro *jFileChooser* que proporciona la interfaz de usuario para seleccionar el archivo que se desea cargar en la herramienta.

Para la implementación de la ayuda centralizada, se ha utilizado la librería JavaHelp tal y como se ha mencionado en las herramientas utilizadas en el capítulo de Estado del arte.

Para ello, en primer lugar se ha instalado la librería en la aplicación. Se ha definido un directorio llamado Help en el que se han introducido todos los ficheros necesarios:

- **Ficheros HTML:** Estos ficheros son los que contienen cada una de las páginas de ayuda de la herramienta que se quiere mostrar. Debe haber un fichero HTML por cada página de ayuda se tenga.
- **Fichero mapa de JavaHelp:** Es un fichero con extensión .jhm (*Java Help Map*) en el que se da una ‘clave’ a cada uno de los ficheros HTML de ayuda que se tienen.
- **Fichero de tabla de contenidos de JavaHelp:** Este fichero tiene extensión .xml y define la tabla de contenidos que aparece en la primera pestañas de parte izquierda de la ventana de ayuda.
- **Fichero índice de JavaHelp:** Este fichero con extensión .xml define un índice alfabético por temas que se encuentra ubicado en la segunda pestaña de la parte izquierda de la ventana de ayuda.
- **Busqueda en la ayuda de JavaHelp:** Esta funcionalidad se encuentra en la tercera pestaña de la parte izquierda de la ventana de ayuda. Para ello, se ha ejecutado una herramienta llamada jhindexer.jar que revisa los ficheros HTML contenidos en la ayuda y genera una pequeña base de datos que sirve para hacer búsquedas más rápidas y eficientes.

- **Posibilidad de introducir duraciones y números no limitados a unos determinados valores.**

En todas las pestañas en las en la versión 1.0 de CourseEditor se introducía duraciones, números de créditos, etc (pestaña “Duración”, “Objetivos” y “UD”) se ha eliminado la restricción de tener que elegir un número determinado que se mostraba al usuario en un *jComboBox*.

Esto se ha conseguido mediante el cambio de todos los *jComboBox* por *TextField*, además de modificar la forma de leer la información de ellos y hacer las comprobaciones necesarias para asegurarse de que el usuario está introduciendo un valor numérico.

- **Introducción de calendarios**

Del mismo modo que ocurría para las duraciones, en los apartados para introducir fechas, concretamente en la pestaña “Duración” para introducir la fecha de inicio y fin del curso definido, se ha implementado un *jDateChooser* utilizando la librería JCalendar, la cual permite implementar calendarios en los que el usuario puede seleccionar las fechas deseadas.

- **Nueva pestaña para la definición de Roles**

En la versión anterior de la herramienta, la definición de los roles que van desempeñar las tareas definidas, se realizaba cuando se introducía más información a cada tarea definida.

En la nueva versión se ha implementado una nueva pestaña llamada “Roles”. La principal novedad que incluye esta pestaña, es que además de asignar una pertenencia (learning o staff) al rol, el usuario tiene la posibilidad de asignar una etiqueta o caracterización (por ejemplo, si el rol es un grupo de prácticas, o un grupo de delegados, etc.) a cada rol.

Para ello, se ha definido una nueva clase llamada *NuevaEtiquetaRol* que implementa la ventana mostrada en el capítulo de Desarrollo del proyecto para la definición de una nueva etiqueta.

#### 4.1.2 Clases *ModeloTablaObjetivos*, *ModeloTablaRoles*, *ModeloTablaTareas* y *ModeloTablaTipoTareas*

Estas clases, las cuales definen las tablas que se muestran en las distintas pestañas de la herramienta, han sido modificadas para introducir botones de ‘Borrar’ y ‘Editar’ en cada ítem.

Para definir cada uno de estos botones se han creado otras dos clases por cada una de ellas que definen el botón y su funcionalidad.

- Para la clase *ModeloTablaObjetivos* se han definido las nuevas clases *BotonTabla* que define el botón de eliminar un objetivo, y *BotonTablaEditar* que define el botón de editar el objetivo.
- Igualmente para la clase *ModeloTablaRoles* se han definido *BotonTablaRoles* y *BotonTablaRolesEditar*.
- Para la clase *ModeloTablaTareas* se han definido *BotonTablaTareas* y *BotoTablaTareasEditar*.
- Finalmente, para la clase *ModeloTablaTipoTareas* se han definido las clase *BotonTabaTipoTareas* y *BotonTablaTipoTareasEditar*.

Por otra parte, en los apartados de las tablas en los que no se puede visualizar correctamente la información porque es demasiado larga, se ha implementado una funcionalidad de tooltips o descripciones emergentes.

Para ello se ha implementado el método:

```
public String getToolTipText(MouseEvent event)
```

### 4.1.3 Clase InfTareaPorSerTarea

Esta clase define la ventana que el usuario debe rellenar para introducir más información a las tareas definidas en la pestaña ‘Tareas’. La clase se ha remodelado completamente de forma que ahora la ventana que se muestra en forma de las pantallas de mostradas consecutivamente, de forma que a medida que se rellena cada pantalla se pulsa el botón ‘Siguiente’ (o ‘Finalizar’ si es la última pantalla) y de esta forma se pasa a la siguiente pantalla.

El resto de cambios más significativos de esta clase incuyen la introducción de calendarios emergentes para la introducción de fechas y la introducción de una ventana extra para la definición de opciones avanzadas que no son necesarias de rellenar, pero el usuario tiene la posibilidad de hacerlo. Esta ventana se ha implementado en la clase nueva introducida llamada *OpcionesAvanzadas1*.

### 4.1.4 Clase EditorGráfico

Esta clase, la cual define el editor gráfico que permite al usuario secuenciar las tareas dentro de cada unidad didáctica, es otra de las clases, junto con *PantallaInicial*, que ha sufrido grandes modificaciones.

En primer lugar, se ha cambiado la interfaz, eliminando muchos de los botones que contenía, así como cambiando el aspecto de las cajas de tipos de tareas. Para ello se ha modificado el método `public Map createCellAttributes(Point2D point, String nombre)`, el cual define las características de las celdas y se ha creado un nuevo directorio llamado *Images* que contiene todos los iconos empleados en todas y cada una de las cajas.

En segundo lugar, se han introducido condiciones que están incluidas en el nivel B de la especificación de e-learning.

Para ello se han modificado y ampliado diversos métodos para contemplar los nuevos casos que se puede dar:

- `public void connect(Port source, Port target, Point2D firstPort, Point2D secondPort)`. Este método es el encargado de controlar las unions que se realizan en el editor, y de comprobar que dichas unions son válidas, ya que si no lo fueran, se muestra un mensaje de error que informa al usuario de ello. Ahora, se ha de tener en cuenta un nuevo tipo de cajas que puede unirse a las ya existentes. Además, las condiciones evalúan propiedades tales como la nota sacada en una tarea o si se ha entregado o no una actividad. Por ello, la caja anterior a una caja unión siempre tiene que ser una tarea de tipo evaluable o entregable. Si no es así, también se ha añadido una comprobación de forma que no se permite al usuario dicha unión y le muestra el mensaje de error correspondiente.

- **public void** borrar (Object c). Este método es el encargado de borrar las celdas y flechas en caso de que el usuario se equivoque al efectuar la secuenciación. Dicho método ha sido modificado para que se permita el borrado de esta nueva celda. Así mismo, se ha modificado para que el resto de celdas correspondientes a los distintos tipos de tarea no se puedan borrar, ya que si se borrara quedaría una tarea que fue definida en la pestaña de la pantalla inicial sin su correspondiente celda. De esta forma, solo está permitido borrar las cajas de Unión y Condición y las flechas.

Por otra parte, se ha creado dos nuevas clases llamadas *EditarCondicionEditor* y *CondicionEditor*. *EditarCondicionEditor* es la encargada de mostrar la ventana que permitirá al usuario definir qué actividad se realizará dependiendo de que condición se cumpla. La clase *CondicionEditor* define los atributos que poseerá una condición del editor gráfico, de forma que nos permita unir distintos atributos en un mismo elemento. Los atributos definidos en esta clase son:

- *String unidad*: Unidad didáctica a la que pertenece la condición
- *String nombre*: nombre de la condición
- *String si*: condición que se debe de cumplir
- *String entonces*: tarea que se realizará si se cumple la condición
- *String sino*: tarea que se realizará si no se cumple la condición
- *int nota*: nota que el alumnos ha obtenido en una actividad evaluable
- *boolean entregable*: boolean que define si la actividad es entregable o no.
- *boolean evaluable*: boolean que define si la actividad es evaluable o no.

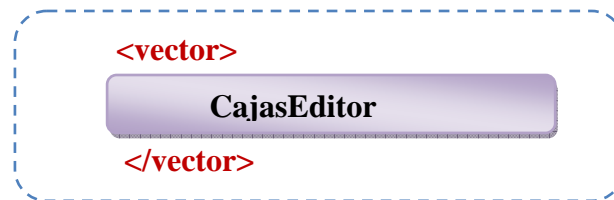
Esta clase cuenta con un constructor y los métodos *get* y *set* correspondientes.

Finalmente se ha añadido una nueva funcionalidad que permite que el editor gráfico sea capaz de guardar la secuenciación de las unidades didácticas de forma que luego se pueda reconstruir en una segunda ejecución.

Para ello, se ha añadido el botón ‘Establecer secuenciación de UD’. El funcionamiento de este botón es el siguiente: una vez acabada la secuenciación de la unidad didáctica actual, si se pulsa el botón mencionado, se crearán dos vectores que contendrán la información de las cajas y de las flechas utilizadas. Gracias a la librería Xstream utilizada, la cual permite serializar objetos a formato XML y viceversa, el contenido de estos vectores son volcados en un fichero XML. Este fichero XML se creará como un archivo temporal en la carpeta de archivos temporales del ordenador del usuario, y posteriormente, al elegir la opción “Guardar” del menú principal, se volcará en la carpeta de archivos guardados.

## CAPÍTULO: Implementación del proyecto

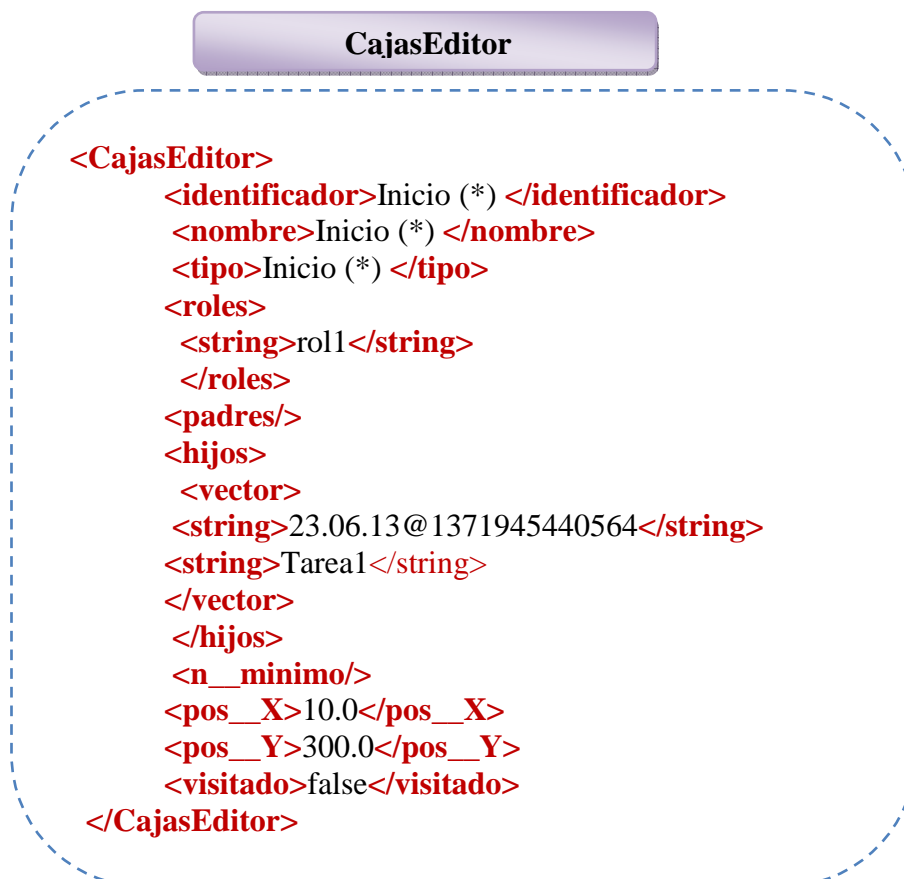
Se creará un fichero XML por cada unidad didáctica con mismo nombre que la unidad didáctica en cuestión. A continuación se puede observar un ejemplo de un fichero XML de una posible secuenciación:



Dentro de *CajasEditor* se incluye la información relativa a cada caja que viene definida por los atributos de la clase del mismo nombre.

- *String identificador*: Identificador asignado a la caja en cuestión.
- *String nombre*: Nombre de la caja
- *String tipo*: Tipo de tarea que define la caja.
- *Vector roles*: Roles que intervienen en la tarea que define la caja
- *Vector padres*: El/los nodos padre/s de la caja en la secuenciación
- *Vector hijos*: El/los nodos hijo/s de la caja en la secuenciación
- *String pos\_X*: posición de la caja en el eje de las X
- *String pos\_Y*: posición de la caja en el eje de las Y
- *boolean visitado*: Boolean que indica si esa caja ha sido o no visitada para su reconstrucción.

A continuación se observa la información de CajasEditor contenida en el fichero XML.





Cuando se ejecuta el editor gráfico para una unidad didáctica ya secuenciada, se lee el XML y se extrae la información que permite volver a reconstruir el gráfico. Para su reconstrucción, se ha utilizado la teoría de grafos y pilas, de forma que se recorren todos los nodos del grafo y de esta forma se reconstruye. Para ello se ha creado la nueva clase *PilaGrafo*.

### 4.1.5 Clase MensajeSalir

Esta clase ha sido añadida para la implementación de una ventana de seguridad que aparecerá si el usuario cierra la aplicación por error sin haber guardado antes el curso que estaba editando.

La ventana preguntará al usuario si desea realizar los cambios guardados en el curso. El usuario tiene tres opciones:

- Pulsar “*Si*”, de forma que aparecerá la ventana que permite al usuario elegir la ruta en la que guardar la documentación.
- Pulsar “*No*”, de forma que la herramienta se cerrará sin guardar nada.
- Pulsar “*Cancelar*”, de forma que desaparecerá la ventana de seguridad y se permitirá al usuario que siga con la definición de su curso.

### 4.1.6 Clases MensajeErrores VectorErrores

Esta clase define la ventana que lista los errores que se han cometido a lo largo de la introducción de información en la definición de un curso tal y como se ha explicado en el apartado ‘*Detección de errores producidos en la introducción de un curso*’ del capítulo de Desarrollo del proyecto.

Para ello, se han creado las clases *VectorErrores* y *MensajeErrores*.

La clase *VectorErrores* contiene los métodos que permiten el manejo y manipulación del vector que contine datos de tipo String con los errores que se van produciendo, así como un método que muestra los errores en una lista en la ventana.

La clase *MensajeErrores* define la interfaz de la ventana de errores que se muestra, así como su funcionalidad.



# Capitulo 5

## Evaluación de la herramienta.

La prueba de un programa es una actividad que forma parte del proceso de desarrollo. Se ha de llevar a cabo según las reglas de la garantía de la calidad y comienza una vez que la actividad de programación se ha finalizado. El objetivo de la realización de pruebas en el software es minimizar las probabilidades de aparición de una anomalía o error en el mismo.

Así pues, una vez finalizada la implementación de la aplicación se ha llevado a cabo una evaluación de la misma mediante una serie de pruebas tanto del código fuente como del comportamiento de la aplicación. Se ha prestado una especial atención y dedicación en la fase de pruebas para asegurar que la versión final está libre de errores y se trata de una versión estable.

Las técnicas de prueba que se han llevado a cabo para determinar el correcto funcionamiento de la aplicación son las siguientes:

- **Pruebas funcionales**

Estas pruebas están orientadas a la ejecución del programa, es decir, a su comportamiento funcional. No se examina el contenido del programa, sino que a partir de la especificación y el ejecutable de la aplicación se utilizan unos determinados datos de prueba que permiten comprobar si se cumplen las especificaciones o no.

- **Pruebas estructurales**

Las técnicas de prueba estructurales están orientadas a detectar errores en el código fuente de la aplicación. Para distintas funcionalidades de la aplicación que se verán a continuación ha sido necesario llevar a cabo pruebas unitarias en las que se verifique el correcto funcionamiento del código de la aplicación. Este tipo de pruebas se han llevado a cabo utilizando el *framework* para la realización de pruebas unitarias sobre código desarrollado en Java, JUnit.

## 5.1 Pruebas funcionales

Para realizar estas pruebas, se ha utilizado la herramienta para crear el diseño de un curso basado en un curso real de una asignatura de grado como el explicado en la Introducción del capítulo Desarrollo del Proyecto.

Estas pruebas se han llevado a cabo para comprobar que la introducción de información en la herramienta se hace de manera correcta, y si no se hace, si el mensaje de error informativo es el correcto.

La estructura de este apartado será la siguiente: por cada pestaña se han realizado pruebas de introducción de datos que se muestran en las siguientes tablas, las cuales incluyen la prueba realizada, los resultados obtenidos y lo que se ha llevado a cabo para solucionar el problema que hubiera si éste seguía existiendo.

➤ **Pestaña Info. General**

Prueba realizada	Resultado obtenido
Se rellena toda la información solicitada en la tabla correctamente.	Al pulsar el botón “Guardar” (o “Guardar como”) no se obtienen mensajes de error.
No se rellena toda la información pedida en la primera pestaña.	Al pulsar el botón “Guardar” (o “Guardar como”) se obtiene el mensaje de error: “Debe rellenar todos los campos en la pestaña Info.General”

*Tabla 6 Pruebas funcionales de la pestaña Info.General de CourseEditor v2.0*

Las dos pruebas realizadas en este apartado se han llevado a cabo con éxito, aún así, se ha echado en falta una funcionalidad que ha sido implementada. El hecho de mostrar este mensaje de error únicamente, ayuda al usuario a saber que hay algún campo que no ha introducido información (o ha introducido información errónea) pero no le ayuda a detectar rápidamente dónde se ha producido el error, sino que el usuario tiene que buscar el campo en el que se produjo dicho error.

Por ello se decidió introducir otra mejora consistente en que al realizar las comprobaciones de cada uno de los campos, el campo que no pase la comprobación de

forma satisfactoria, su *jLabel* correspondiente será coloreado de rojo. Esta mejora se ha implementado para la información introducida en todas las pestañas de la herramienta.

Se ha llevado a cabo pues, más pruebas que corroboren el correcto funcionamiento de esta mejora. En la Tabla 7 se encuentra recogida:

Prueba realizada	Resultado obtenido
Se han llevado a cabo distintas posibilidades en las que no se rellenen uno ó más campos de la pestaña.	Se ha comprobado que los <i>jLabel</i> se han coloreado correctamente.

Tabla 7 Pruebas funcionales de la pestaña Info.General de CourseEditor v2.0

Por otro lado se han comprobado las opciones del botón de tabulación para que un usuario si lo desea pueda ir pasando por todos los campos sin necesidad de utilizar el ratón.

Prueba realizada	Resultado obtenido
Se ha probado que la opción de tabulador funcionara en los distintos campos de la pestaña.	<p>Al realizar la prueba se ha obtenido que dicho botón solo realizaba su acción de forma correcta cuando se encontraba el cursor en un <i>jTextField</i>, es decir, de la parte de Título a Descripción. Como para la introducción de información del resto de campos se encuentra definido un <i>jTextArea</i> ya que los datos a introducir pueden ser más amplios, la opción tabulador no funcionaba.</p> <p>Esto ocurría porque un <i>jTextArea</i> es un componente que permite al usuario introducir multiples líneas de texto y por lo tanto también admite opciones de tabulado, por lo que si se presiona este botón, en lugar de cambiar el foco al siguiente componente se introduce una tabulación en el mismo campo. Para solucionar este problema, se ha añadido un <i>KeyListener</i> a cada uno de los <i>jTextArea</i> existentes. La función que tiene el <i>KeyListener</i> es la de tranferir el foco al siguiente componente cuando se pulse el botón de tabulación.</p>

Tabla 8 Pruebas funcionales de la pestaña Info.General de CourseEditor v2.0

### ➤ Pestaña Duración

Para comprobar que el usuario está introduciendo correctamente la información, se llevarán a cabo las comprobaciones recogidas en la Tabla 9:

Prueba realizada	Resultado obtenido
Comprobación de que el número de horas del curso introducido sea un valor numérico.	Se ha comprobado que la lectura y comprobación de la validez del número de horas introducido han sido satisfactorias.
Comprobación de que la fecha de fin introducida no es posterior a la fecha de inicio	Se ha comprobado que la comprobación de que las fechas introducidas fueran correctas ha resultado satisfactorio.  No obstante, después de la realización de las pruebas, se ha llegado a la conclusión de que existía un caso en el que la comprobación fallaba. Este caso se daba cuando la fecha de inicio y de fin eran la misma, ya que podría ser posible que un curso durara algunas horas y comenzara y acabara en el mismo día. Por lo tanto se ha cambiado la comprobación de fechas y se ha añadido la posibilidad de que esto ocurra.
Comprobación de que el número de créditos introducidos sea un valor numérico	Se ha comprobado que la lectura y comprobación de la validez del número de créditos introducido han sido satisfactorias.

*Tabla 9 Pruebas funcionales de la pestaña Duración de CourseEditor v2.0*

Finalmente, al igual que en el apartado anterior se han realizado varias pruebas para comprobar que el funcionamiento del coloreado de los JLabel en los que se ha producido algún error se ha llevado a cabo con éxito.

### ➤ Pestaña Objetivos

Las comprobaciones llevadas a cabo se recogen en la Tabla 10:

Prueba realizada	Resultado obtenido
Comprobación de que todos los campos necesarios para definir un objetivo hayan sido rellenados.	Al pulsar el botón “Añadir”, si falta algún campo sin rellenar, aparece una ventana informativa que avisa al usuario de que no puede establecer un objetivo si no ha rellenado todos los apartados.
Comprobación de que el valor del peso sea un número decimal no superior a 1.	Se ha modificado en la herramienta la forma de obtener el valor introducido, ya que antes el usuario disponía de un

	<p><i>jComboBox</i> para seleccionar el valor deseado y ahora introduce el número por teclado en un <i>jTextField</i>. Se ha probado que la comprobación se realiza de manera correcta, es decir, el usuario no puede introducir números enteros mayores que 1 ni caracteres.</p> <p>No obstante, se ha detectado que para que la herramienta considere como correcto un número decimal introducido, el decimal tiene que estar representado por un punto y no por una coma (por ejemplo, 0.5) Por lo tanto, esta información también ha sido introducida en el mensaje de error que le aparece al usuario.</p>
Comprobación de que la suma de todos los pesos no sea superior a 1.	Se ha comprobado que la suma de los pesos introducidos es la correcta y que si el valor a introducir sumado a los que ya había introducidos va a ser superior a 1, aparece un mensaje de error que informa al usuario.
Comprobación de que al pulsar el botón editar, la información se carga correctamente en los campos de introducción de información y permite editar y volver a introducir la información editada en la tabla.	<p>La forma elegida para editar una tarea ha sido la de recoger la información de la tarea editar, sacarla de la tabla y cargarla de nuevo en los campos de introducción de información de la parte de arriba. Esta forma ha sido elegida ya que evitaba la aparición de otra ventana auxiliar y por lo tanto es más cómoda para el usuario.</p> <p>No obstante, al realizar la prueba, apareció un nuevo problema, que consistía en que una vez que el creador de cursos hubiera editado la información y volviera a pulsar el botón “Añadir” para volver a introducir de nuevo la información en la tabla, ésta se introducía al final de la tabla, de forma que los objetivos no quedaban en la misma posición en la que estaban inicialmente.</p> <p>Esto se ha solucionado de forma que cuando un usuario pulsa el botón “Editar”, no solo se recupera la información a editar, sino también el número de fila en el que se encontraba dicho objetivo, de forma que al introducir de nuevo el objetivo con la información editada, se introduzca en la misma posición en la que se encontraba previamente.</p>

Comprobación de que al pulsar el botón “Borrar” de un objetivo, este se elimina correctamente.	Se ha comprobado que en el caso de borrar un objetivo, no existía ningún problema y el objetivo deseado se elimina de forma correcta.
--	---

*Tabla 10 Pruebas funcionales de la pestaña Objetivos de CourseEditor v2.0*

### ➤ Pestaña U.D

Las comprobaciones que se llevarán a cabo se muestran en la Tabla 11.

Prueba realizada	Resultado obtenido
Comprobación de que todos los campos necesarios para definir una unidad didáctica hayan sido rellenados.	Al igual que en la pestaña anterior se ha comprobado que en el caso de que uno de los campos quedara incompleto, aparece un mensaje de error informando al usuario cuando presiona el botón “Añadir”
Comprobación de que el número de minutos introducidos sea un valor numérico.	Se ha comprobado que la nueva implementación para controlar que el valor introducido no contiene caracteres no numéricos se ha realizado de forma correcta, es decir, si el usuario introduce la información de manera errónea, aparecerá una ventana de error informando del error.
Comprobación de que al pulsar el botón editar, la información se carga correctamente en los campos de introducción de información y permite editar y volver a introducir la información editada en la tabla	De la misma forma que en apartado anterior, se han llevado a cabo pruebas para comprobar el correcto funcionamiento del botón “Editar” y se ha introducido la mejora de recordar la fila de la que fue extraída la unidad didáctica a editar.
Comprobación de que al pulsar el botón “Borrar, la unidad didáctica se elimina correctamente.	Igual que en el apartado anterior se ha realizado esta comprobación sin ningún problema a reseñar.

*Tabla 11 Pruebas funcionales de la pestaña U.D de CourseEditor v2.0*

### ➤ Pestaña T.Tareas

Las pruebas que se han llevado a cabo en esta pestaña están recogidas en la Tabla 12.

Prueba realizada	Resultado obtenido
Comprobación del correcto funcionamiento del botón “Editar” de cada uno de los tipos de tareas.	Se ha probado que se realiza correctamente la carga de la información contenida en el tipo de tarea a editar en la ventana emergente Tipo Tarea.
Comprobación del correcto funcionamiento del botón “Borrar” de cada	Al igual que en las pestañas anteriores se ha probado que el borrado de ítems se



uno de los tipos de tareas.	realiza de forma correcta.
Comprobación de la creación de un nuevo tipo de tarea definido por el usuario.	<p>Cuando el usuario crea un nuevo tipo de tarea, tiene que introducir una información obligatoria. Se ha probado que si no se introduce toda la información requerida aparece correctamente la ventana de error que informa de ello.</p> <p>Además, al añadir la posibilidad de que el usuario escoja un icono que represente al nuevo tipo de tarea, ha surgido la necesidad de crear una nueva comprobación que verifique que el icono introducido por el usuario cumple con los requisitos de tipo y tamaño requeridos. Para ello se ha probado a introducir varios tipos de iconos, y se ha comprobado que cuando el icono introducido no cumple las características, aparece un mensaje de error informando al usuario de ello.</p>

*Tabla 12 Pruebas funcionales de la pestaña T.Tareas de CourseEditor v2.0*

### ➤ Pestaña Roles

Las comprobaciones que se han llevado a cabo en esta pestaña se pueden observar en la Tabla 13:

Prueba realizada	Resultado obtenido
Comprobación de que todos los campos estén rellenos antes de pulsar el botón “Añadir rol”	Al igual que en el resto de pestañas se ha probado que se realice correctamente la comprobación de la información introducida.
Comprobación del correcto funcionamiento de la acción de crear una nueva etiqueta.	<p>Se ha probado que las etiquetas definidas por el usuario se crean correctamente y se incorporan al <i>jComboBox</i> de etiquetas para que el usuario tenga la posibilidad de reutilizarla en posteriores roles definidos.</p> <p>Por otra parte, al realizar dichas pruebas, se pensó que podría surgir la necesidad de que un usuario necesitara borrar una etiqueta previamente creada en el caso de que se hubiera equivocado al definirla. Por lo tanto, se ha introducido un nuevo botón que permite borrar la etiqueta seleccionada en el <i>jComboBox</i>.</p>
Comprobación del correcto	Al igual que en las pestañas anteriores,

funcionamiento del botón “Editar” de cada uno de los roles	surgió el mismo problema de que al editar un determinado rol e introducirlo de nuevo en la tabla, estos se desordenaban. Una vez subsanado este error se ha comprobado que el botón funciona correctamente.
Comprobación del correcto funcionamiento del botón “Borrar” de cada uno de los roles	Al igual que en los apartados anteriores, se ha probado que el funcionamiento de este botón es correcto.

*Tabla 13 Pruebas funcionales de la pestaña Roles de CourseEditor v2.0*

### ➤ Pestaña Tareas

Las comprobaciones que se han realizado se pueden observar en la Tabla 14.

<b>Prueba realizada</b>	<b>Resultado obtenido</b>
Comprobación de que todos los campos necesarios para la definición de una nueva tarea son correctos.	<p>Se han realizado pruebas para verificar que la información introducida es la correcta.</p> <p>En el caso del segundo campo, Duración, ha sido necesario introducir una comprobación adicional que permita controlar que lo que el usuario ha introducido sea un valor numérico.</p> <p>En el caso del tercer campo, Tipo, se comprueba correctamente que existe al menos un tipo de tarea creado en la pestaña de tipos de tareas.</p> <p>En el último campo, U.D, se comprueba correctamente que existe al menos una unidad didáctica creada en la pestaña correspondiente.</p> <p>Para la realización de esta prueba se han introducido diferentes valores, válidos y no válidos, que han permitido comprobar que los mensajes de error obtenidos son los correctos.</p>
Comprobación del correcto funcionamiento del botón “Editar” de cada uno de los roles	<p>Se ha probado que al pulsar el botón “Editar” de distintas tareas, la información de las mismas se han cargado correctamente en la ventana emergente de introducción de más información.</p> <p>Aquí, al igual que en los apartados</p>

	anteriores surgió el problema de desordenación de los elementos en la tabla, por lo que también se han introducido cambios para controlarlo.
Comprobación del correcto funcionamiento del botón “Borrar” de cada uno de los roles	Al igual que en los apartados anteriores, se ha probado que el funcionamiento de este botón es correcto.

*Tabla 14 Pruebas funcionales realizadas en la pestaña Tareas de CourseEditor v2.0*

### ➤ Pestaña UD-Tareas

Las comprobación realizada en la pestaña UD-Tareas se recoge en la Tabla 15.

Prueba realizada	Resultado obtenido
Comprobación de que todas las unidades didácticas hayan sido secuenciadas.	Para ello, en el curso definido se ha probado a no secuenciar ninguna de las unidades didácticas, a secuenciar solo alguna de ellas y finalmente a secuenciar todas.  El resultado obtenido ha sido el correcto ya que mientras quedara alguna unidad didáctica sin secuenciar se ha obtenido el error al pulsar “Guardar” o “Guardar como” del menú desplegable.

*Tabla 15 Pruebas funcionales de la pestaña UD-Tareas de CourseEditor v2.0*

### ➤ Editor gráfico

Las pruebas llevadas a cabo se encuentran reflejadas en la Tabla 16.

Prueba realizada	Resultado obtenido
Comprobación de uniones.  Para estas comprobaciones se han realizado diferentes pruebas: <ul style="list-style-type: none"> <li>- Intento de unir una caja con otra que no ha sido previamente editada.</li> <li>- Intento de unir una caja correspondiente a una tarea con dos cajas hijas.</li> </ul>	En todos los casos el resultado obtenido ha sido el esperado. El editor no permite al usuario trazar la flecha errónea y al mismo tiempo aparece el mensaje de error informativo correspondiente.

<ul style="list-style-type: none"> <li>- Intento de trazado de flechas recíprocas.</li> <li>- Intento de unir una tarea que no es ni entregable ni evaluable con una caja de condición.</li> <li>- La secuenciación de una unidad didáctica debe acabar en una única tarea.</li> </ul>	
<p>Comprobación del funcionamiento de borrado de elementos en el editor gráfico.</p>	<p>En la versión 1.0 de la herramienta se permitía el borrado tanto de cajas como de flechas. En la versión actual, aparecen las cajas de cada una de las tareas definidas por el usuario de manera automática sin que el usuario tenga que introducirlas manualmente al editor. Esto, ha creado una nueva necesidad, ya que si se permite el borrado de una caja, se perdería una tarea que está definida en la tabla correspondiente de la herramienta, por lo tanto, se ha decidido que hay que introducir un cambio de funcionalidad de forma que no se permita el borrado de cajas pertenecientes a tipos de tareas. Por lo tanto, solo se permite borrar las uniones y las cajas de Condición y Unión.</p> <p>Una vez realizados estos cambios, se ha probado de nuevo que el funcionamiento es el esperado.</p>
<p>Comprobación del guardado de la secuenciación de cada una de las unidades didácticas.</p>	<p>Se ha comprobado que los ficheros XML que permitirán la posterior recuperación de la secuenciación, se corresponden con los esperados y se crean correctamente dentro de la carpeta comprimida en formato zip que contiene toda la documentación generada por la herramienta.</p>
<p>Comprobación del establecimiento de condiciones. Para poder editar una caja de condición se debe cumplir que la caja padre pertenezca a una tarea evaluable o entregable y que además la condición tenga dos tareas hijas.</p>	<p>Se ha probado que cuando no se cumplen los requisitos que permiten editar la condición, ésta no puede ser editada y aparecen los mensajes de error correspondientes.</p> <p>Una vez editada se ha comprobado que la información se guarda correctamente y se incorpora tanto en el fichero de texto que guarda toda la información del curso, como en el fichero XML que</p>

sigue la especificación IMS-LD.
---------------------------------

*Tabla 16 Pruebas funcionales del editor gráfico de CourseEditor v2.0*

## 5.2 Pruebas estructurales

Estas pruebas se han llevado a cabo para comprobar que la carga de información desde un fichero en la herramienta se realiza de manera correcta.

Para ello se ha creado una clase que cumple los siguientes criterios:

- Es una subclase de `TestCase`
- Tiene tantos métodos como se quiera probar funcionalidad del código de la aplicación. Todos ellos tienen que empezar por la palabra `test`. Estos métodos compararán el valor que esperamos recibir con el valor devuelto, de forma que si ambos son iguales, el test se habrá pasado con éxito. Para la realización de estas pruebas en concreto se ha realizado un método que puede ser reutilizado cambiando la información por cada pestaña, excepto para la pestaña de Duración que se han necesitado definir cuatro métodos para probar toda la funcionalidad.
- Un método `main()` que permite ejecutar el test y presentar su salida.

Para la evaluación, se han llevado a cabo comprobaciones sobre la carga desde fichero de la información de todas las pestañas:

### ➤ Pestaña Info. General

Se ha comprobado que la información de cada uno de los espacios de introducción de texto, se ha cargado exactamente de la misma forma que estaba guardado en fichero

La comprobación que se realiza con este método se recoge en la Tabla 17.

Valor de entrada guardado	Valor esperado
Programación de Sistemas	Programación de Sistemas

*Tabla 17 Tabla de valor esperado de las pruebas estructurales en la pestaña Info.General de CourseEditor v2.0*

El método definido para ello:

```
public void testInfoGeneral() {
    assertEquals("Programación de Sistemas",
        PantallaInicial.jTextField1.getText());
}
```

Para comprobar el resto de campos de texto, solo habría que cambiar `jTextField1` por el `jTextField` deseado.

El resultado obtenido se puede observar en la Figura 81.

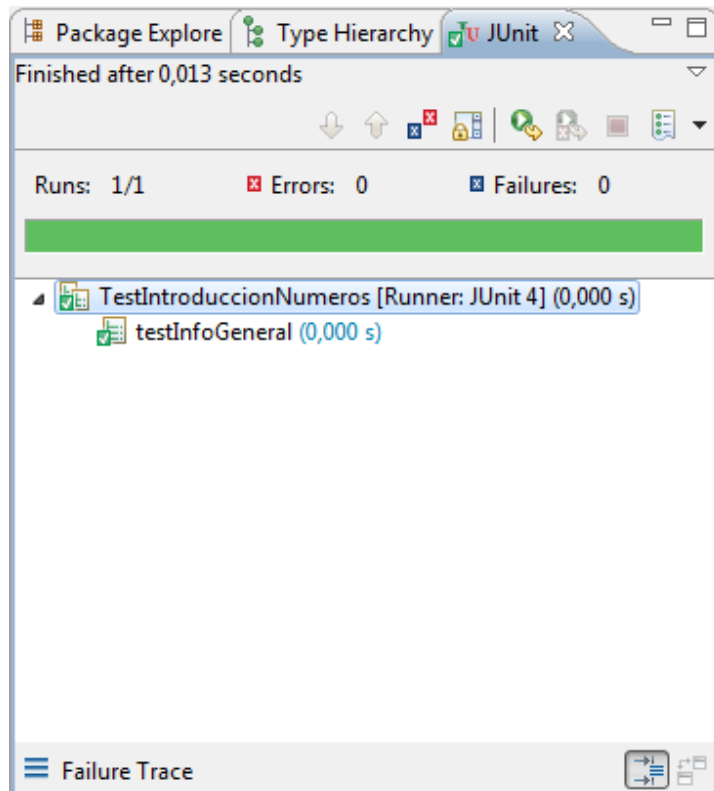


Figura 81 Resultado de la prueba unitaria de la pestaña Info. General

#### ➤ Pestaña Duración.

En esta pestaña se ha comprobado que la información de cada uno de los espacios de introducción de texto para número de horas y número de créditos, se ha cargado exactamente de la misma forma que estaba guardado en fichero. También se ha comprobado que las fechas de inicio y fin del curso se cargan correctamente en los calendarios.

- Comprobación del número de horas del curso:

Valor de entrada guardado	Valor esperado
42	42

Tabla 18 Número de horas esperado. Pruebas estructurales en la pestaña Duración de CourseEditor v2.0

El método implementado para ello es:

```
public void testNumeroHoras() {
    assertEquals("42", PantallaInicial.jTextField7.getText());
}
```

- Comprobación de la fecha de inicio del curso:

Valor de entrada guardado	Valor esperado
28/1/2013	28/1/2013

*Tabla 19 Fecha de inicio esperada. Pruebas estructurales en la pestaña Duración de CourseEditor v2.0*

El método implementado para ello es:

```
public void testFechaInicio() {
    String diP =
String.valueOf(PantallaInicial.jCalendar1.getCalendar().get(Calendar.DAY_OF_MONTH));
    String miP =
String.valueOf(PantallaInicial.jCalendar1.getCalendar().get(Calendar.MONTH)+1);
    String aiP =
String.valueOf(PantallaInicial.jCalendar1.getCalendar().get(Calendar.YEAR));

    assertEquals("28/1/2013", diP+"/"+miP+"/"+aiP);
}
```

- Comprobación de la fecha de fin del curso:

Valor de entrada guardado	Valor esperado
10/5/2013	10/05/2013

*Tabla 20 Fecha de fin esperada. Pruebas estructurales en la pestaña Duración de CourseEditor v2.0*

El método implementado para ello es:

```
public void testFechaFin() {
    String diF =
String.valueOf(PantallaInicial.jCalendar2.getCalendar().get(Calendar.DAY_OF_MONTH));
    String miF =
String.valueOf(PantallaInicial.jCalendar2.getCalendar().get(Calendar.MONTH)+1);
    String aiF =
String.valueOf(PantallaInicial.jCalendar2.getCalendar().get(Calendar.YEAR));

    assertEquals("10/5/2013", diF+"/"+miF+"/"+aiF);
}
```

- Comprobación del número de créditos:

Valor de entrada guardado	Valor esperado
6	6

Tabla 21 Número de créditos esperado. Pruebas estructurales en la pestaña Duración de CourseEditor v2.0

El método implementado para ello es:

```
public void testNumeroCreditos() {  
    assertEquals("6", PantallaInicial.jTextField8.getText());  
}
```

Los resultados obtenidos se pueden observar en la Figura 82.

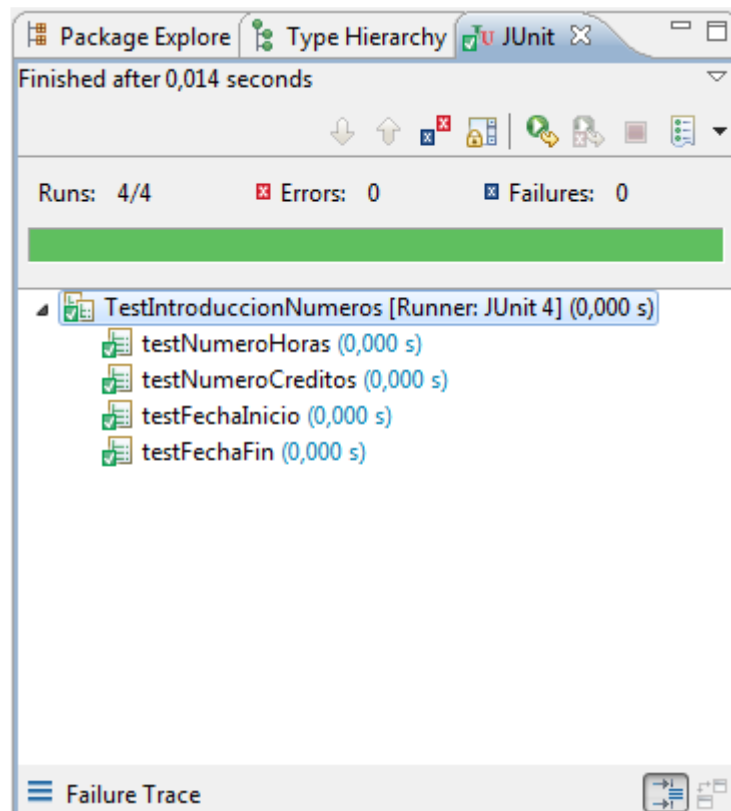


Figura 82 Resultado de las pruebas unitarias de la pestaña Duración



### ➤ Pestaña Objetivos

En esta pestaña se ha comprobado que se introduce correctamente la información cargada desde fichero en la tabla.

Valor de entrada guardado	Valor esperado
0.1	0.1

Tabla 22 Peso esperado. Pruebas estructurales en la pestaña Objetivos de CourseEditor v2.0

El método de test creado para ello es el siguiente.

```
public void testTablaObjetivos() {
    String peso = "0.1";
    assertEquals(peso, (String)PantallaInicial.mod.getValueAt(1,1));
}
```

Para comprobar el resto de campos de texto, solo habría que cambiar el número de fila y columna que se pasa como parámetro en el método `getValue` (`int numFila`, `int numColumna`), y cambiar el valor esperado por el que corresponda.

Los resultados obtenidos se pueden observar en la Figura 83.

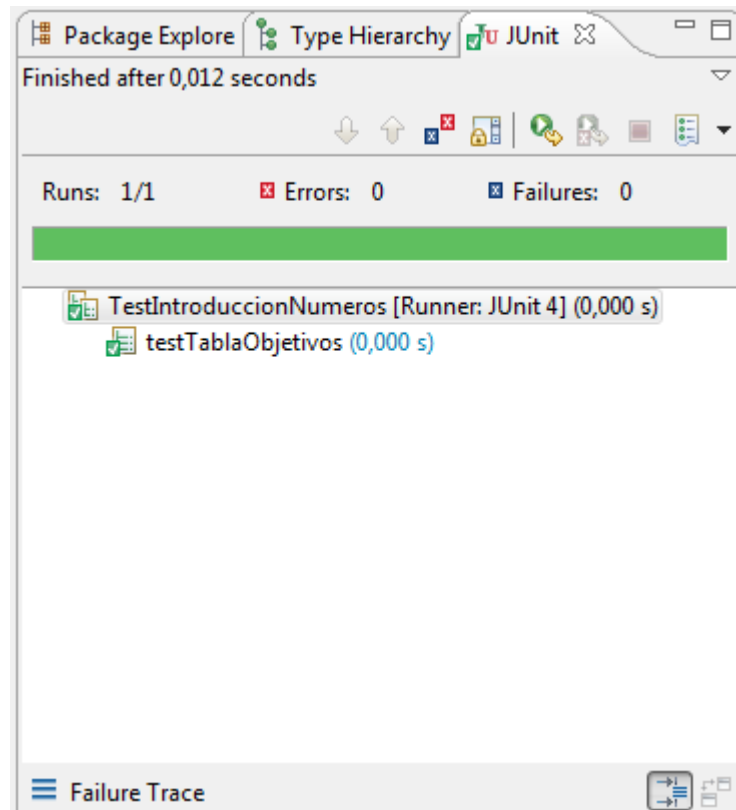


Figura 83 Resultado de las pruebas unitarias de la pestaña Objetivos

### ➤ Pestaña U.D

En esta pestaña se ha comprobado que se introduce correctamente la información cargada desde fichero en la tabla.

Valor de entrada guardado	Valor esperado
90	90

Tabla 23 Valor de la tabla de U.D esperado. Pruebas estructurales en la pestaña U.D de CourseEditor v2.0

El método de test creado para ello es el siguiente:

```
public void testTablaUnidades() {
    String duracion = "90";
    assertEquals(duracion, (String)PantallaInicial.mod2.getValueAt(0,2));
}
```

Para comprobar el resto de campos de texto, solo habría que cambiar el número de fila y columna que se pasa como parámetro en el método `getValue` (`int numFila`, `int numColumna`), y cambiar el valor esperado por el que corresponda.

Los resultados obtenidos se pueden observar en la Figura 84.

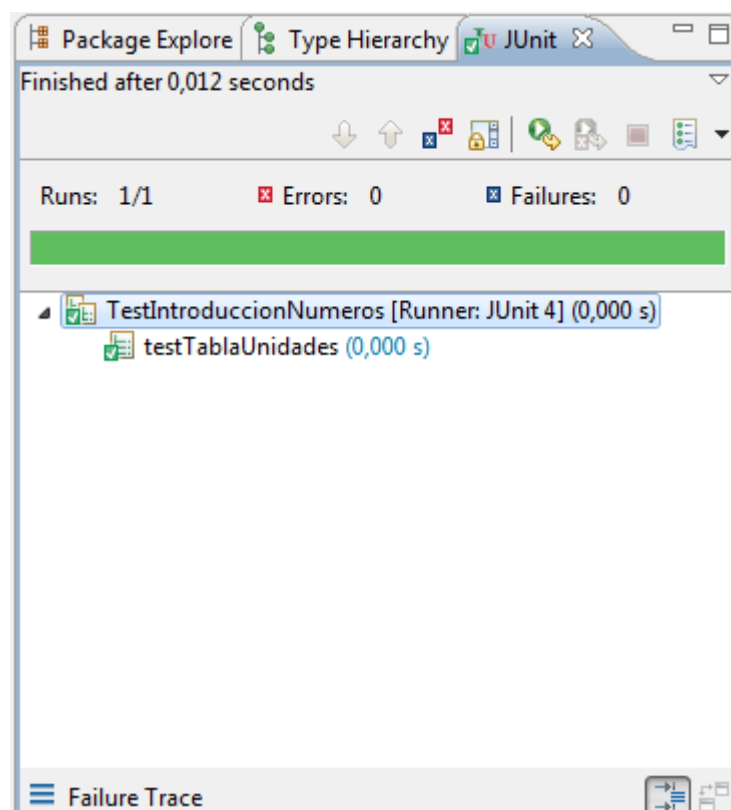


Figura 84 Resultado de las pruebas unitarias de la pestaña U.D

#### ➤ Pestaña T. Tareas

En esta pestaña se ha comprobado que se introduce correctamente la información cargada desde fichero en la tabla.

Valor de entrada guardado	Valor esperado
Chat	Chat

Tabla 24 Tabla del valor de la tabla T.Tareas esperado. Pruebas estructurales en la pestaña T.Tareas de CourseEditor v2.0

El método de test creado para ello es el siguiente.

```
public void testTablaTipoTareas() {
    String nombreTipo = "Chat";
    assertEquals(nombreTipo, (String)PantallaInicial.mod3.getValueAt(0,0));
}
```

Para comprobar el resto de campos de texto, solo habría que cambiar el número de fila y columna que se pasa como parámetro en el método `getValue` (`int numFila`, `int numColumna`), y cambiar el valor esperado por el que corresponda.

Los resultados obtenidos se pueden observar en la Figura 85.

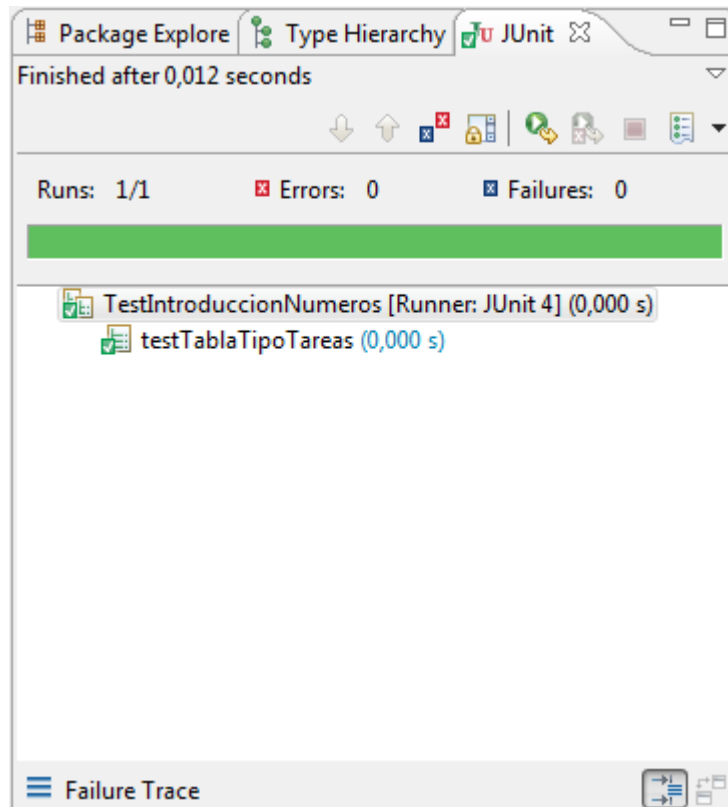


Figura 85 Resultado de las pruebas unitarias de la pestaña T.Tareas

### ➤ Pestaña Roles

En esta pestaña se ha comprobado que se introduce correctamente la información cargada desde fichero en la tabla.

Valor de entrada guardado	Valor esperado
staff	staff

Tabla 25 Valor de la tabla de roles esperado. Pruebas estructurales en la pestaña Roles de CourseEditor v2.0

El método de test creado para ello es el siguiente.

```
public void testTablaRoles() {
    String pertenencia = "staff";
    assertEquals(pertenencia, (String)PantallaInicial.mod5.getValueAt(0,2));
}
```

Para comprobar el resto de campos de texto, solo habría que cambiar el número de fila y columna que se pasa como parámetro en el método `getValue` (`int numFila`, `int numColumna`), y cambiar el valor esperado por el que corresponda.

Los resultados obtenidos se pueden observar en la Figura 86.

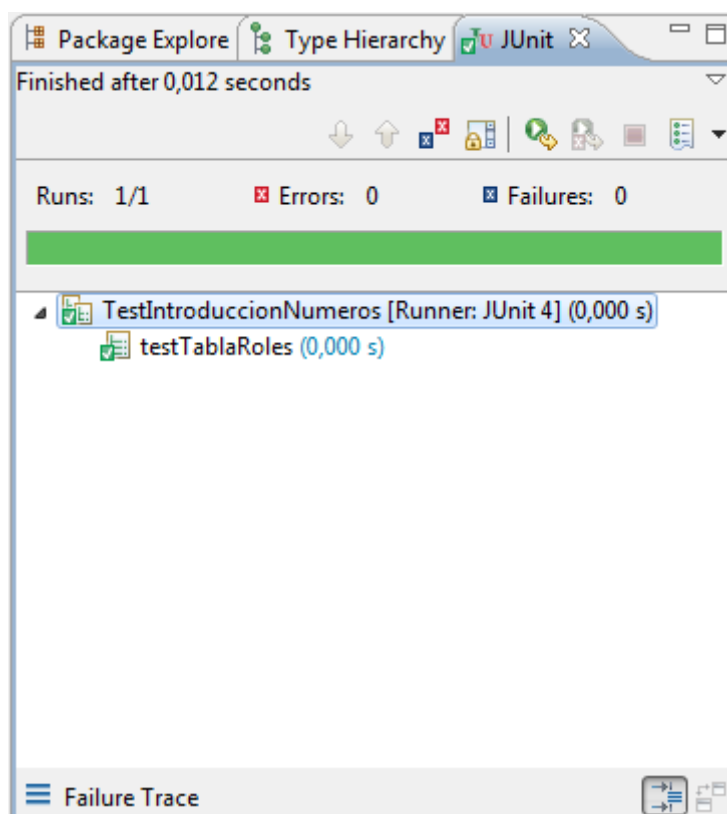


Figura 86 Resultado de las pruebas unitarias de la pestaña Roles

### ➤ Pestaña Tareas

En esta pestaña se ha comprobado que se introduce correctamente la información cargada desde fichero en la tabla.

Valor de entrada guardado	Valor esperado
17.01.13@1358420122986	17.01.13@1358420122986

Tabla 26 Valor de la tabla de tareas esperado. Pruebas estructurales en la pestaña Tareas de CourseEditor v2.0

El método de test creado para ello es el siguiente.

```
public void testTablaTareas() {
    String id = "17.01.13@1358420122986";
    assertEquals(id, (String)PantallaInicial.mod4.getValueAt(0,5));
}
```

Para comprobar el resto de campos de texto, solo habría que cambiar el número de fila y columna que se pasa como parámetro en el método `getValue` (`int numFila`, `int numColumna`), y cambiar el valor esperado por el que corresponda.

Los resultados obtenidos se pueden observar en la Figura 87.

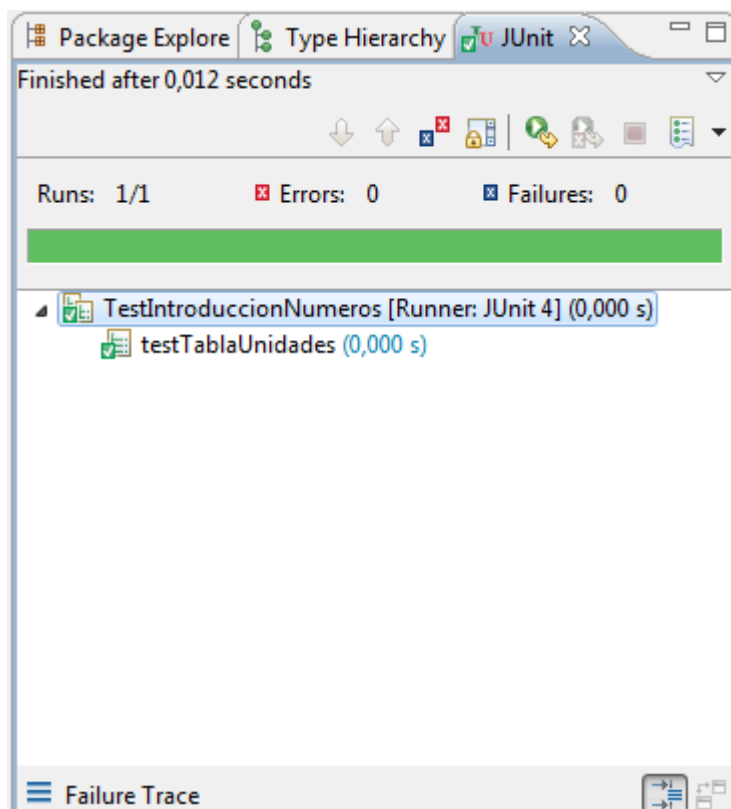


Figura 87 Resultado de las pruebas unitarias de la pestaña Tareas

## 5.3 Secuenciaciones realizadas

Por último, se han realizado pruebas con las secuenciaciones de cada una de las unidades didácticas del curso de prueba creado con el objetivo de determinar y solucionar posibles fallos en las mismas.

Estas unidades didácticas se han cargado en un reproductor de IMS-LD llamado Coppercore player. A continuación se muestran los pasos a seguir para cargar un curso en el player.

CopperCore Run Time environment (CCRT) disponible tanto para Windows como para Linux, se puede descargar desde: <http://coppercore.sourceforge.net/downloads.shtml>.

Una vez descomprimida la carpeta, debemos ejecutar el fichero *coppercore.bat*, de forma que se ejecuta una nueva consola, la cual contiene el servidor de aplicación JBoss.

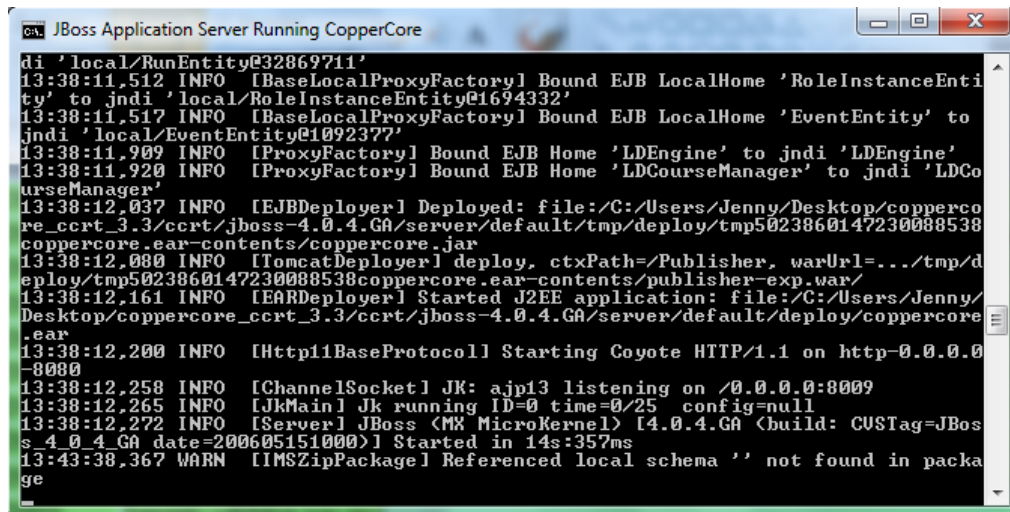


Figura 88 Ejecución del script CopperCore.bat

Después, se debe acceder a la interfaz de CopperCore desde la siguiente URL:  
<http://localhost:8080/Publisher/publication.html>



Figura 89 Interfaz de CopperCore player.

A continuación se debe seleccionar la carpeta comprimida en .zip que contiene los ficheros necesarios para la creación del curso y pulsar en la opción “Publish”. Si la validación resulta satisfactoria aparecerá una pantalla como la siguiente:

## CAPÍTULO: Evaluación de la herramienta.

Time	Level	Message
0	INFO	Validation started.
0	INFO	step 1 - Analysing package (C:\Users\Jenny\Desktop\coppercore_ccrt_3.3\ccrt\data/upload\insmanifest.zip).
10	INFO	step 2 - validating the manifest.
50	INFO	step 3 - validating global content.
50	INFO	step 4 - checking if all files in package are referenced.
50	INFO	Validation passed successfully.
50	INFO	Start processing manifest
50	INFO	Processing manifest started
210	INFO	Successfully build component model
210	INFO	Semantic validation was successful
290	INFO	Processing manifest succeeded
290	INFO	Storing local webresources in C:\Users\Jenny\Desktop\coppercore_ccrt_3.3\ccrt\jboss-4.0.4.GA\server\default\deploy\jbossweb-tomcat55.sar\ROOT.war\1\
300	INFO	Resources are stored.

Figura 90 Mensaje obtenido tras una validación con éxito.

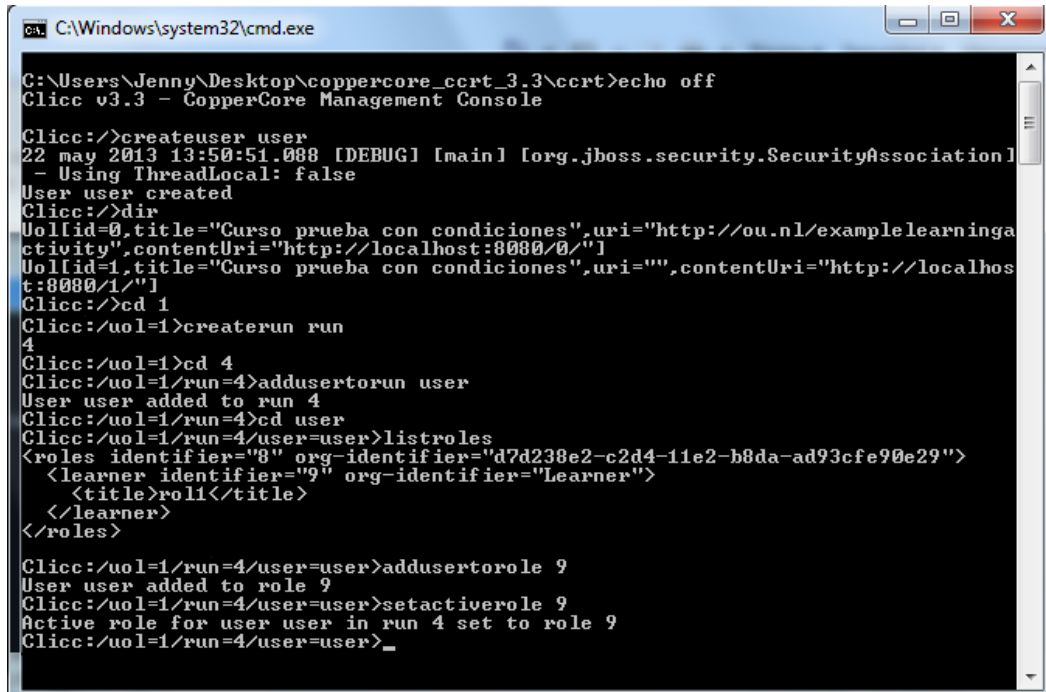
Una vez realizada con éxito la publicación del curso, mediante el uso de la interfaz Clicc de CopperCore, se crean los alumnos y tutores así como la instancia del curso. Para ello se han de seguir los siguientes pasos:

1. Se crea un nuevo usuario mediante el comando:  
*Clicc:/>createuser <id del usuario>*
2. Se obtiene la lista de los paquetes IMS-LD publicados mediante el comando:  
*Clicc:/>dir*
3. Se selecciona la publicación que deseamos utilizar para visualizar su demo mediante el comando:  
*Clicc:/>cd <id de la publicación>*
4. Se crea una nueva ejecución mediante el commando:  
*Clicc:/uol=1>createrun <título de la ejecución>*
5. Se selecciona el identificador de la ejecución que deseamos mediante el comando:  
*Clicc:/uol=1>cd <id de la ejecución>*
6. Se añade un usuario a la ejecución mediante el comando:  
*Clicc:/uol=1>addusertorun <userid of the user>*
7. Se selecciona el usuario deseado mediante el comando:  
*Clicc:/uol=1/run=0>cd <id del usuario>*
8. Se obtiene la lista de roles definidos en el paquete IMS-LD mediante el comando. En el caso de prueba llevado a cabo se obtendrían los todos los roles definidos para el curso (alumnos, profesor de teoría y profesor de prácticas)  
*Clicc:/uol=1/run=0/user=fred>listroles*
9. Se añade un usuario a cada rol. Este proceso se repetirá todos y cada uno de los roles a los que necesitemos añadir un usuario.  
*Clicc:/uol=1/run=0/user=fred>addusertorole <identifier of the role>*
10. Se define uno de los roles como 'rol activo' para la ejecución mediante el comando de forma que la visualización del curso se hará desde el punto de vista del rol que hemos definido como activo.  
*Clicc:/uol=1/run=0/user=fred>setactiverole <identifier of the active role>*



Si se desean asignar más usuarios a la ejecución, se ha de volver a repetir desde el paso 6 hasta el final todas las veces que sea necesario.

Todos estos pasos se pueden visualizar en la Figura 91.



```

C:\Windows\system32\cmd.exe

C:\Users\Jenny\Desktop\coppercore_ccrt_3.3\ccrt>echo off
Click v3.3 - CopperCore Management Console

Click:/>createuser user
22 may 2013 13:50:51.088 [DEBUG] [main] [org.jboss.security.SecurityAssociation]
- Using ThreadLocal: false
User user created
Click:/>dir
Uollid=0,title="Curso prueba con condiciones",uri="http://ou.nl/examplelearninga
ctivity",contentUri="http://localhost:8080/0/"]
Uollid=1,title="Curso prueba con condiciones",uri="",contentUri="http://localhos
t:8080/1/"
Click:/>cd 1
Click:/uol=1>createrun run
4
Click:/uol=1>cd 4
Click:/uol=1/run=4>addusertorun user
User user added to run 4
Click:/uol=1/run=4>cd user
Click:/uol=1/run=4/user=user>listroles
<roles identifier="8" org-identifier="d7d238e2-c2d4-11e2-b8da-ad93cfe90e29">
  <learner identifier="9" org-identifier="Learner">
    <title>rol1</title>
  </learner>
</roles>
Click:/uol=1/run=4/user=user>addusertorole 9
User user added to role 9
Click:/uol=1/run=4/user=user>setactiverole 9
Active role for user user in run 4 set to role 9
Click:/uol=1/run=4/user=user>_
  
```

Figura 91 Pasos realizados para la ejecución del Click

Una vez realizados los pasos descritos anteriormente es posible comenzar la ejecución del *reproductor* principal. Para ello habrá que acceder a la siguiente URL en local: <http://localhost:8080/WebPlayer/runswitch.html>

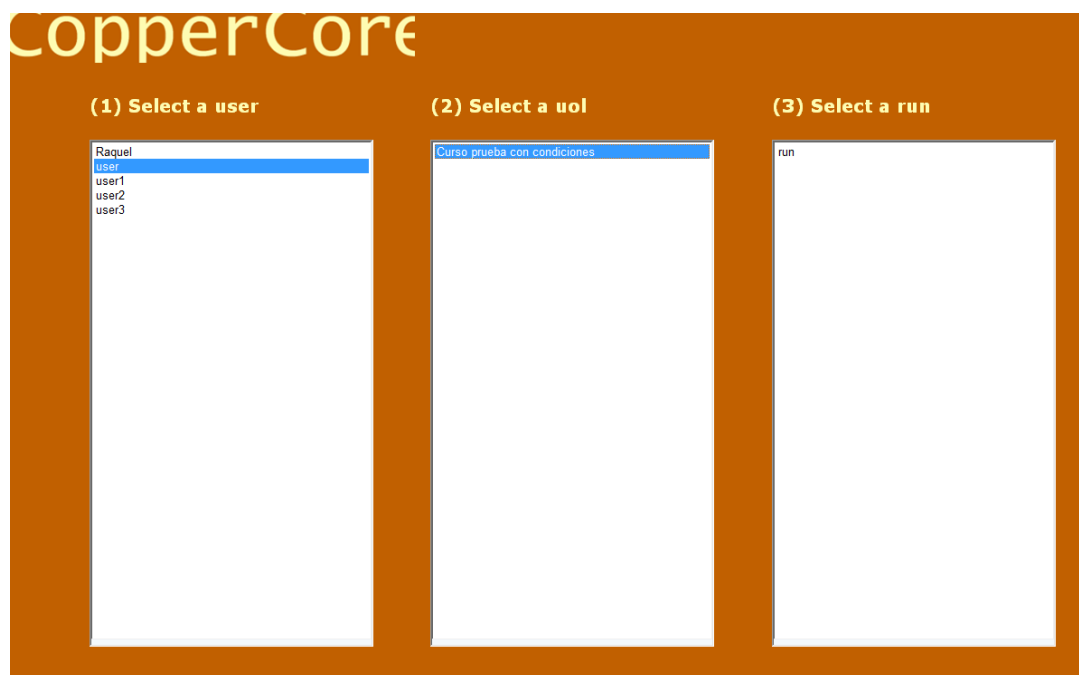


Figura 92 Interfaz de CopperCore player

Desde aquí, se seleccionara el usuario, la unidad de aprendizaje y la ejecución que se desea visualizar en el *reproductor*.

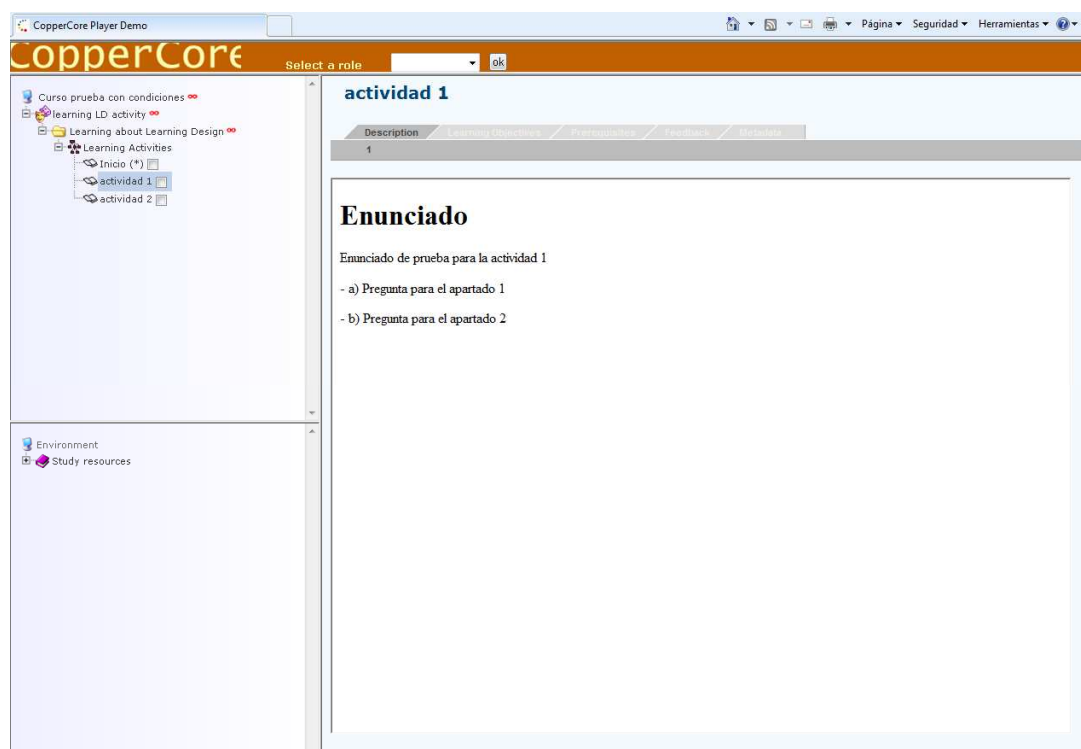


Figura 93 Visualización de la ejecución de la unidad de aprendizaje en el reproductor.

En el anexo 2 al final de esta memoria se puede ver en detalle el enunciado del curso creado y en el anexo 3 las secuenciaciones que se han realizado.

# Capítulo 6

## Gestión del proyecto

En este capítulo se presenta la metodología llevada a cabo en la realización del presente proyecto de fin de carrera. En primer lugar, se describen las fases de desarrollo del proyecto en las que se especifican las tareas llevadas a cabo en cada fase. A continuación, se presenta la planificación temporal del proyecto mediante un diagrama de Gantt. Y por último, se incluye el presupuesto del presente proyecto en el que se detalla el desglose de costes así como el coste final que supone el desarrollo del proyecto.

### 6.1 Fases del desarrollo

El presente trabajo se ha llevado a cabo en diferentes etapas atendiendo a las fases de desarrollo que siguen los sistemas de software del mercado que se desarrollan actualmente. Dichas fases son la de planificación, desarrollo y documentación.

A continuación se detallan las fases citadas anteriormente.

#### 6.1.1 Planificación

En las primeras etapas del proyecto se definieron los objetivos generales y el funcionamiento deseado para la aplicación. Para ello se definió un listado inicial de objetivos y especificaciones que la nueva versión de CourseEditor debía cumplir.

Adicionalmente, se creó una cuenta de usuario en la herramienta Bugzilla para reportar los avances o errores encontrados en la herramienta en todas las fases de desarrollo del proyecto.

La planificación del desarrollo del proyecto se ha llevado a cabo mediante el modelo de trabajo para la gestión y desarrollo de software denominado *Scrum*. Dentro de esta metodología ha tenido especial importancia en el desarrollo de este proyecto la realización de *sprints*. En estos periodos se llevaba a cabo la realización de las tareas específicas del proyecto y tras cada *sprint* se planificaba una reunión en la que se verificaban los avances realizados en la herramienta y se definían nuevos objetivos o se adaptaban los existentes en función de las necesidades que surgían. Al finalizar cada reunión, se redactaba un acta de reunión en el que se detallaban las acciones llevadas a cabo hasta el momento y se indicaban las acciones a realizar durante el siguiente periodo de *sprint*.

Además, en esta fase se llevó a cabo un estudio de las tecnologías necesarias para llevar a cabo la implementación de la aplicación, así como una búsqueda exhaustiva de documentación relativa a las citadas tecnologías.

### 6.1.2 Desarrollo

La fase de desarrollo se compone de distintas subfases que se describen a continuación:

- **Análisis y diseño**

En esta fase se ha efectuado un análisis de las distintas funcionalidades en los diferentes módulos y submódulos del sistema. Se ha realizado una evaluación de la versión inicial de CourseEditor con el objetivo de detectar los errores existentes y documentarlos para asignarles una prioridad de resolución en las reuniones periódicas realizadas.

- **Desarrollo**

Esta fase engloba la implementación de todos los módulos y submódulos de la aplicación. De acuerdo a la planificación del proyecto, se han realizado diecisiete *sprints* en los que se llevaba a cabo la implementación de los cambios y mejoras definidos en el listado de especificaciones de la nueva versión de CourseEditor. Al final de cada *sprint* y antes de cada reunión se realizaba una copia del ejecutable de la aplicación en el repositorio SVN en el cual se encuentra la misma.

- **Evaluación**

Para realizar la evaluación de la herramienta se han llevado a cabo pruebas tanto funcionales como estructurales.

En esta fase, se ha creado un curso utilizando la herramienta tomando como referencia el programa de la asignatura ‘Programación de sistemas’ de grado en Ingeniería Telemática.

Además, se han realizado diferentes secuenciaciones utilizando el editor gráfico de la herramienta para detectar fallos y verificar el correcto funcionamiento de diferentes casos de prueba. Así pues, se han considerado tanto casos de prueba en los que el resultado esperado debía ser exitoso como casos de prueba en los que se esperaba un error.

Se ha prestado una especial atención y dedicación en la fase de pruebas para asegurar que la versión final se trata de una versión estable.

### 6.1.3 Documentación

Durante el desarrollo de proyecto se ha ido documentando la información que se ha considerado importante para el mismo. En la herramienta Bugzilla se han documentado todos los errores encontrados en la versión inicial de la herramienta y todos los errores corregidos en la nueva versión. También, se ha llevado a cabo un histórico de todas las actas redactadas tras las reuniones y se han realizado distintas copias de la herramienta en sus diferentes estados a lo largo del desarrollo del proyecto en el repositorio habilitado para ello.

Finalmente se ha redactado el presente documento de memoria del proyecto final de carrera y se ha preparado la presentación.

## 6.2 Planificación temporal

Una vez establecidos los objetivos del proyecto y las fases de desarrollo del mismo se ha realizado la planificación temporal con la ayuda de un diagrama de Gantt. Este tipo de diagrama muestra el tiempo de dedicación previsto para cada tarea a lo largo del tiempo previsto para la realización del proyecto.

Al comienzo y finalización del proyecto se han establecido dos hitos indicando la fecha de inicio y fin del mismo. Del mismo modo, se han establecido hitos que marcan las reuniones que han tenido lugar y que marcan el comienzo y fin de cada *sprint*. Los hitos son tareas de duración nula que se utilizan para marcar puntos de control en el transcurso del proyecto.

El diagrama de Gantt resultante se puede observar en la Figura 94 . En este diagrama se muestra una visión global de la planificación del proyecto, de sus distintas fases y de su duración.

CAPÍTULO: Gestión del proyecto

Inicio del Proyecto	0 días	jue 01/03/12	jue 01/03/12
Fase de Planificación	102 días	jue 01/03/12	vie 20/07/12
Reunión	0 días	jue 01/03/12	jue 01/03/12
Definición de objetivos	22 días	jue 01/03/12	vie 30/03/12
Reunión	0 días	jue 22/03/12	jue 22/03/12
Reunión	0 días	mié 28/03/12	mié 28/03/12
Reunión	0 días	jue 19/04/12	jue 19/04/12
Análisis de funcionalidad de CourseEditor	18 días	vie 30/03/12	mar 24/04/12
Documentación	23 días	mar 24/04/12	jue 24/05/12
Reunión	0 días	jue 03/05/12	jue 03/05/12
Reunión	0 días	jue 10/05/12	jue 10/05/12
Estudio de errores existentes en CourseEditor	39 días	jue 24/05/12	mar 17/07/12
Reunión	0 días	jue 31/05/12	jue 31/05/12
Documentación y priorización de errores en Bugzilla	39 días	jue 24/05/12	mar 17/07/12
Reunión	0 días	jue 28/06/12	jue 28/06/12
Fase de Desarrollo	185 días	mar 17/07/12	lun 01/04/13
Reunión	0 días	mié 18/07/12	mié 18/07/12
Reunión	0 días	jue 06/09/12	jue 06/09/12
Implementación de cambios en la interfaz de la herramienta	65 días	mar 17/07/12	lun 15/10/12
Reunión	0 días	jue 11/10/12	jue 11/10/12
Reunión	0 días	mar 16/10/12	mar 16/10/12
Solución de errores de la herramienta	57 días	mar 16/10/12	mié 02/01/13
Reunión	0 días	jue 24/01/13	jue 24/01/13
Implementación de nuevas funcionalidades	63 días	jue 03/01/13	lun 01/04/13
Fase de pruebas	36 días	mar 02/04/13	mar 21/05/13
Pruebas Funcionales	17 días	mar 02/04/13	mié 24/04/13
Reunión	0 días	mar 16/04/13	mar 16/04/13
Pruebas Estructurales	16 días	jue 25/04/13	jue 16/05/13
Realización de secuenciaciones	9 días	vie 17/05/13	mié 29/05/13
Fase de Documentación	22 días	mié 29/05/13	jue 27/06/13
Reunión	0 días	mar 25/06/13	mar 25/06/13
Redacción de la Memoria	22 días	mié 29/05/13	jue 27/06/13
Presentación	14 días	lun 10/06/13	jue 27/06/13
Fin del Proyecto	1 día	jue 27/06/13	jue 27/06/13

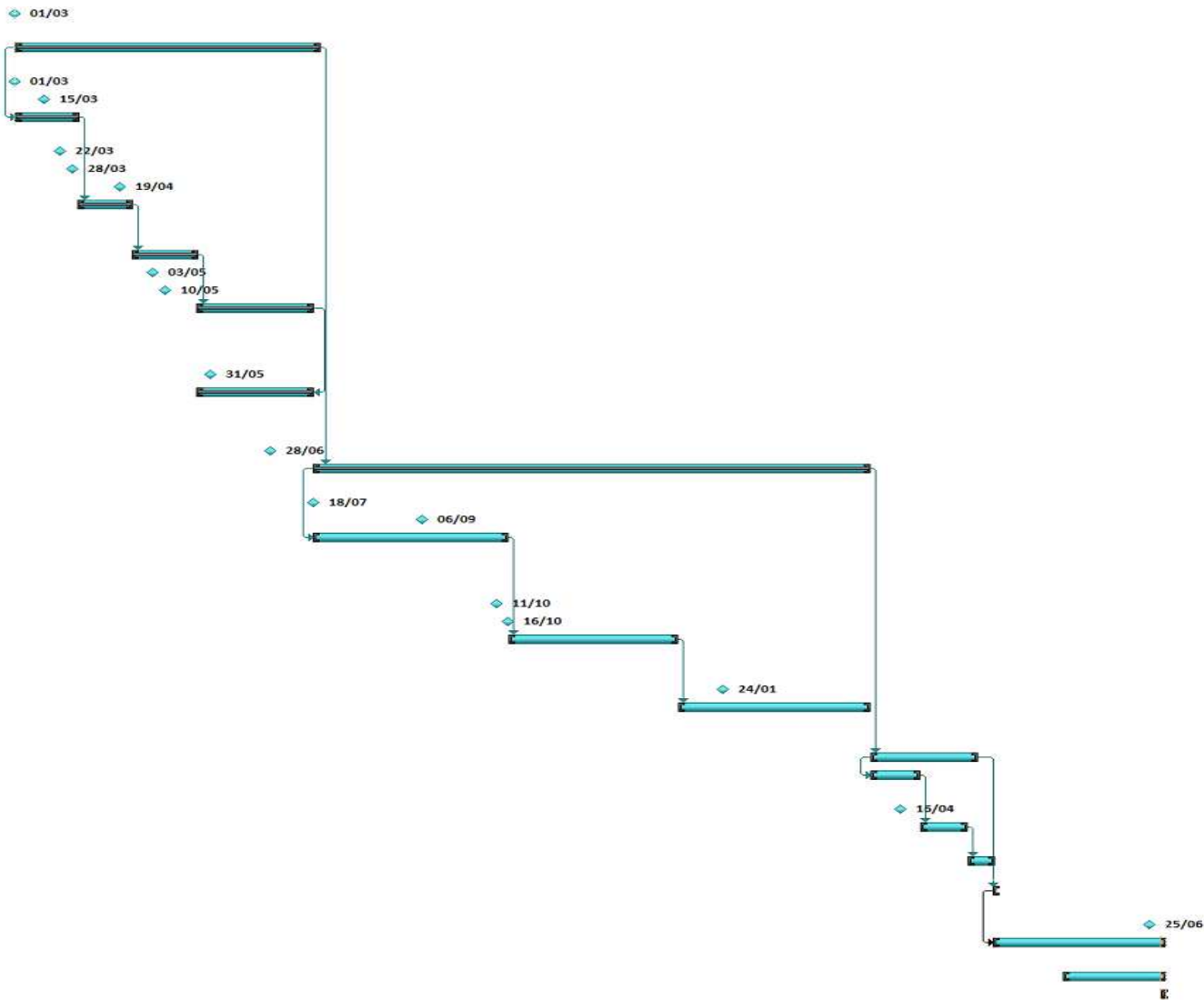


Figura 94 Diagrama de Gantt

## 6.3 Presupuesto

A continuación se presenta el presupuesto del presente proyecto de fin de carrera. Se incluye un desglose de las fases del mismo, así como un desglose de costes de personal y material. Por último se indica el coste total del proyecto.

- **Tareas**

- Fase de planificación

- Definición de objetivos. Duración → 22 días.
- Análisis de funcionalidad de CourseEditor. Duración → 18 días.
- Documentación. Duración → 23 días.
- Estudio de errores existentes en CourseEditor. Duración → 39 días.
- Documentación y priorización de errores en Bugzilla. Duración → 39 días.

- Fase de desarrollo

- Implementación de cambios en la interfaz de la herramienta. Duración → 65 días.
- Solución de errores de la herramienta. Duración → 57 días.
- Implementación de nuevas funcionalidades. Duración → 63 días.

- Fase de pruebas

- Pruebas funcionales. Duración → 17 días.
- Pruebas estructurales. Duración → 16 días.
- Realización de secuenciaciones. Duración → 9 días.

- Fase de documentación

- Redacción de la memoria. Duración → 22 días.
- Realización de la presentación. Duración → 14 días.

- **Recursos**

Los recursos utilizados para la realización del presente proyecto son los que se detallan a continuación.

### ○ Recursos software

- Java. Coste → 0 €
- Plataforma de desarrollo Eclipse. Coste → 0 €
- Bugzilla. Coste → 0 €
- Google Drive. Coste → 0 €
- Licencia de Microsoft Office 2010. Coste → 269 €

### ○ Recursos hardware

- Ordenador Portátil. Coste → 950 €

Para calcular el coste imputable del equipo, se ha supuesto un tiempo de amortización típico de 5 años. El número de horas estimadas al año del funcionamiento del ordenador se ha establecido en 1800 horas. Durante el desarrollo del proyecto, el ordenador ha sido utilizado durante 900 horas, lo cual supone una amortización anual del 50%.

En la Tabla 27 se muestra la cantidad imputable por el uso del hardware.

Elemento	Uds.	Horas de proyecto	Horas por año	Precio	Amortización anual	Coste imputable
<b>Ordenador portátil</b>	1	900	1800	950	50%	95 €

*Tabla 27 Coste de los recursos hardware*

### ○ Recursos humanos

En la realización de este proyecto han participado tres personas, los directores de proyecto y el desarrollador. Las horas dedicadas al proyecto han sido de una media de 4 horas diarias, contemplando semanas de 7 días. El coste de un ingeniero es de 22 €/hora.

En la Tabla 28 se indica el desglose de costes en recursos humanos en cada fase del proyecto.



Fase	Duración	Importe
Fase de planificación	102 días	8.976 €
Fase de desarrollo	185 días	16.280 €
Fase de pruebas	36 días	3.168 €
Fase de documentación	22 días	1.936 €
Directores del proyecto	25 horas	1100 €
<b>Total</b>	345 días y 25 horas	31.460€

*Tabla 28 Desglose de costes en recursos humanos*

Por tanto, el **coste total** del proyecto se detalla en la Tabla 29.

Concepto	Importe
Recursos Software	269 €
Recursos Hardware	95 €
Recursos Humanos	31.460 €
<b>Subtotal</b>	31.824 €
<b>(21% IVA)</b>	6.683,04 €
<b>TOTAL</b>	38.507,04€

*Tabla 29 Coste total del proyecto*

El presupuesto total de este proyecto asciende a la cantidad de **TREINTA Y OCHO MIL QUINIENTOS SIETE EUROS Y CUATRO CENTIMOS.**

Madrid, a 25 de Junio

Fdo. Raquel Catalina Vaquero



# Capítulo 7

## Conclusiones y trabajo futuro

En este capítulo se presentan las principales conclusiones extraídas de la realización del presente proyecto de fin de carrera. A partir de los resultados logrados se comprueba que los objetivos iniciales han sido cumplidos. Por último se hace un análisis de los posibles trabajos futuros que se podrían llevar a cabo en la aplicación para aumentar su funcionalidad y su eficiencia.

### 7.1 Conclusiones

En este proyecto de fin de carrera se ha llevado a cabo una remodelación y mejora de la herramienta llamada CourseEditor. Esta herramienta permite a un profesor crear un curso de manera sencilla, abarcando aspectos relevantes de la programación curricular del mismo.

Todos los objetivos definidos en el apartado 1.2 de esta memoria han sido cumplidos. Asimismo, a lo largo de la implementación del proyecto han ido surgiendo nuevas ideas y se han ido añadiendo nuevos requisitos que se han considerado más apropiados para mejorar aún más la aplicación. El resultado final ha sido una herramienta, que, no solo ha ampliado la facilidad de uso de la versión anterior, sino que también incorpora la funcionalidad contemplada en el nivel B de la especificación IMS-LD.

Se han llevado a cabo diferentes acciones para mejorar sustancialmente la usabilidad de la herramienta. En primer lugar, se ha introducido una funcionalidad de guardado

automático de la información introducida en cada pestaña cada vez que se cambia de la misma. Esta es la mejora más importante dado que en la versión anterior de la herramienta a un usuario le podía resultar bastante tedioso el hecho de tener que pulsar el botón “Aceptar” de cada una de las pestañas antes de pasar a la siguiente, ya que en caso de no hacerlo así, el curso no se crearía correctamente.

Otra mejora importante ha sido la de la introducción de un menú desplegable que incluye las funcionalidades ‘Nuevo’, ‘Abrir’, ‘Guardar’, ‘Guardar como’, ‘Temas de ayuda’ y ‘Acerca de’.

Se han mejorado distintos campos de introducción de información en la herramienta. Por un lado, se han revisado los tamaños de la interfaz principal y las diferentes tablas de las que consta la herramienta. De esta forma se ha conseguido que sea más fácil visualizar la información introducida en las mismas. Por otro lado, se han quitado limitaciones a la hora de introducir valores numéricos y fechas en la herramienta.

Se ha implementado una funcionalidad que comprueba si existen errores en la introducción de información de la herramienta, y muestre una lista al usuario con estos errores antes de guardar el curso creado permitiendo una modificación posterior. Un nuevo requisito que se ha añadido mientras que se implementaba la funcionalidad anterior ha sido la de a la vez que se realicen las comprobaciones necesarias, se coloreen en rojo los campos en los que haya existido algún error en la introducción de información. De esta forma se ha conseguido que sea más fácil detectar los errores cometidos para el creador de cursos.

Otra mejora importante ha sido la de una implementación que permite a los usuarios decidir la ruta en la que quieren guardar el curso creado, el cual se guardará junto con toda la documentación generada por la herramienta en una carpeta comprimida en zip, la cual será compatible con otras aplicaciones que también siguen la especificación IMS-LD. Esto ha dotado a la herramienta de mayor flexibilidad y comodidad, ya que el usuario puede tener una carpeta destinada a sus cursos creados, y de ésta forma, guardarlo ahí directamente, es decir, no tiene que estar moviendo la carpeta manualmente a posteriori.

Se ha habilitado una nueva pestaña de definición de roles más específica en la que se incluyen características de éstos tales como etiquetas que permiten caracterizar al tipo de rol que se está definiendo (por ejemplo, si se trata de un grupo de alumnos de prácticas, de un grupo de delegados, etc.). En la herramienta se han definido de forma predeterminada algunas de ellas, pero también el usuario puede crear y borrar de forma muy sencilla todas las que considere necesarias para el curso.

Se ha llevado a cabo una importante remodelación en el editor gráfico de la herramienta. Entre estos cambios cabe destacar, la disminución del número de botones de la misma lo cual lleva asociado la disminución del número de clicks asociados a cada acción, debido a que ahora todas las tareas definidas anteriormente para cada unidad didáctica saldrán automáticamente en la parte editable del editor. La inclusión de nuevas cajas ‘condición’ para cubrir las necesidades de la nueva funcionalidad añadida comentada a continuación, así como la introducción de iconos asociados a cada tipo, de forma que permite al usuario un mejor reconocimiento de los mismos.

Así mismo, si el usuario define una nueva tarea personalizada, también tiene la posibilidad de añadir un icono personalizado asociada a la misma.

Se han incluido nuevas funcionalidades de nivel B recogidas en la especificación IMS-LD. Concretamente se ha introducido la posibilidad de añadir condiciones que permitirán a un diseñador de cursos decidir que tarea debe o no debe realizar el alumno dependiendo por ejemplo, de la nota que haya sacado en un examen, o de el cumplimiento o no en el tiempo de entrega de un determinado trabajo. Para la realización de esta nueva funcionalidad se ha estudiado a fondo la información del nivel B de la especificación IMS-LD y las condiciones se han definido de manera suficientemente sencilla para que un profesor que no tenga conocimientos técnicos de la especificación ni de e-learning pueda definirlas sin dificultad.

Finalmente, se ha implementado una nueva funcionalidad de guardado de la secuenciación gráfica. De esta forma, el usuario podrá guardar el gráfico generado, y éste será cargado de manera automática en el editor gráfico cada vez que se vuelva a abrir. Toda la información relativa al gráfico se guardará en ficheros XML (uno por cada secuenciación) con el nombre de cada una de las unidades didácticas en la misma ubicación donde se ha decidido guardar el resto de ficheros del curso. Esta ha sido la parte más complicada de la realización del proyecto, ya que se estudiaron distintas formas de llevar a cabo la implementación y se tuvieron en cuenta distintas librerías Java que permitían verter la información de un gráfico a un fichero XML. Finalmente se decidió utilizar Xstream por ser la librería que permitía generar un fichero XML que solo contuviera la información necesaria.

Como conclusión final, los objetivos establecidos al inicio de este proyecto de fin de carrera se han cumplido y teniendo en cuenta todos y cada uno de los cambios realizados y las mejoras introducidas, se puede concluir que este proyecto de fin de carrera se ha llevado a cabo de manera satisfactoria y que ha dado como resultado una herramienta que aumenta la usabilidad existente en la versión anterior, a la vez que permite una mayor funcionalidad a la hora de definir cursos por parte de un profesor.

Personalmente, la realización del presente proyecto de fin de carrera me ha permitido conocer en profundidad las diferentes especificaciones de e-learning existentes, su importancia y las ventajas que añaden a los procesos de aprendizaje. Con respecto a la implementación me ha permitido ampliar mis conocimientos de programación en Java, concretamente, la de la realización de interfaces de usuario utilizando la biblioteca gráfica java.Swing, y trabajar con diferentes librerías de código abierto que permiten añadir mejoras muy útiles a dichas interfaces. También me ha permitido conocer las diferentes fases de desarrollo de un proyecto, así como la evaluación del funcionamiento de cada uno de los métodos de clases Java gracias a la utilización de la librería JUnit.

## 7.2 Trabajo futuro

A continuación se detallan los aspectos sobre los que se podría trabajar para obtener mayores prestaciones en la aplicación desarrollada.

- Inclusión de aspectos no considerados del nivel A de la especificación, como por ejemplo la etiqueta *on-completion* de la especificación que permite especificar la acción o acciones que se realizarán después de que la actividad actual haya sido terminada o permitir establecer un mayor número de posibilidades de sincronía en el editor gráfico para ampliar las posibilidades de definición de cursos y aumentar la flexibilidad que poseerán cada uno de los roles a lo largo de la realización del curso. Actualmente existen tres posibilidades de sincronía: por curso, por unidad didáctica o por tarea.
- Inclusión de aspectos no considerados del nivel B de la especificación. Actualmente se ha introducido la posibilidad de definir condiciones en la secuenciación, de forma que un alumno deba realizar una tarea u otra dependiendo de un aspecto evaluable o entregable de la actividad anterior. No obstante, el nivel B de la especificación también contempla la introducción de propiedades genéricas. Una posible implementación de ellos sería la posibilidad de almacenar información sobre personas (preferencias, resultados, información personal...), sobre un rol o sobre el diseño de aprendizaje en sí mismo. Esto se podría incluir en la nueva pestaña implementada para la definición de roles.
- Otros posibles líneas de trabajo futuro incluirían la inclusión de funcionalidades de nivel C de la especificación, el cual incluye notificaciones, las cuales permiten especificar el envío de mensajes y el establecimiento de nuevas actividades en base a ciertos eventos, por ejemplo, la posibilidad de notificar al profesor mediante correo electrónico de un trabajo entregado por el alumno para ser evaluado o el aumento de nivel de jerarquía de un curso, ya que actualmente en la herramienta el nivel es de tres, es decir, existen unidades didácticas, y dentro de éstas hay tareas y dentro de éstas hay roles y actividades de soporte. Otro posible trabajo futuro consistiría en aumentar este nivel. Por ejemplo, dentro de un rol o actividad soporte, podría haber contenido más información específica de ellos.

# Anexo 1

## Tutorial para la realización de modificaciones en la herramienta.

En este tutorial se explicará cómo realizar cambios en la herramienta CourseEditor de forma que se pueda modificar o ampliar su funcionalidad usando Windows 7 como sistema operativo. En primer lugar, se explicarán los requisitos necesarios que debe cumplir el sistema, a continuación se explicará el proceso de instalación del software necesario y por último se dará una visión general del código de la aplicación y de sus clases y métodos más característicos.

Por tanto, la estructura de este tutorial es la siguiente:

- Instalación de Java JDK
- Instalación de Eclipse
- Uso de la herramienta Eclipse
- Código de la aplicación (meter aquí la forma de obtenerlo -> repositorio!!!!)
- Principales clases y métodos de la aplicación

### *1. Instalación de Java JDK*

Para poder programar en Java y por tanto compilar y ejecutar código escrito en el mencionado lenguaje de programación es necesario disponer de *Java Development Kit* (JDK) instalado en la máquina. JDK es un software que provee herramientas de

CAPÍTULO: Tutorial para la realización de modificaciones en la herramienta.

desarrollo para la creación de programas en Java. Puede ser descargado desde la página web de Oracle [36]

Una vez descargado el archivo, hay que darle doble clic para que empiece la instalación del mismo. Aparecerá una pantalla como la siguiente:

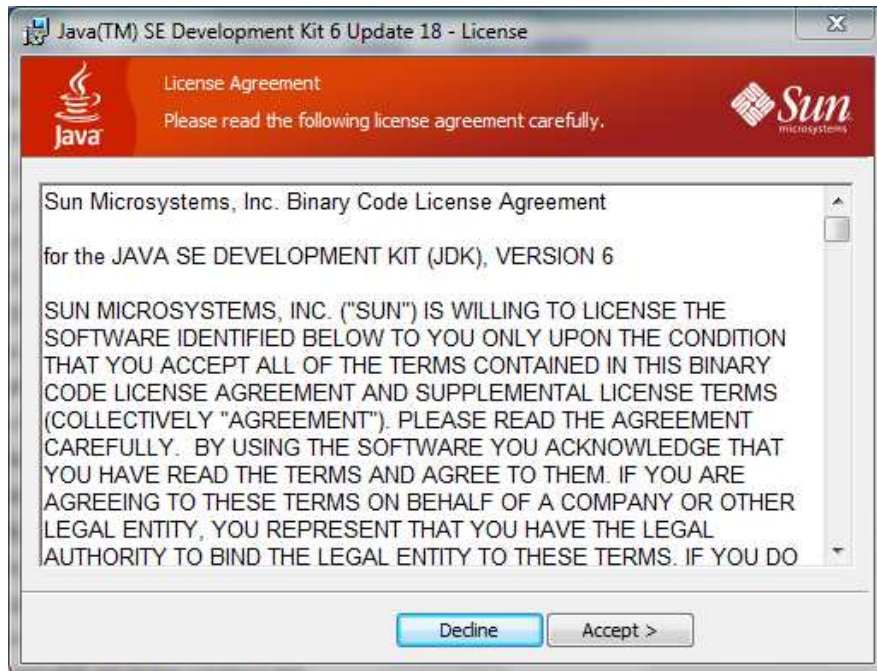


Figura 95. Asistente de instalación de Java Development kit

Una vez leído el contrato de usuario final se ha de pulsar aceptar (Accept >)

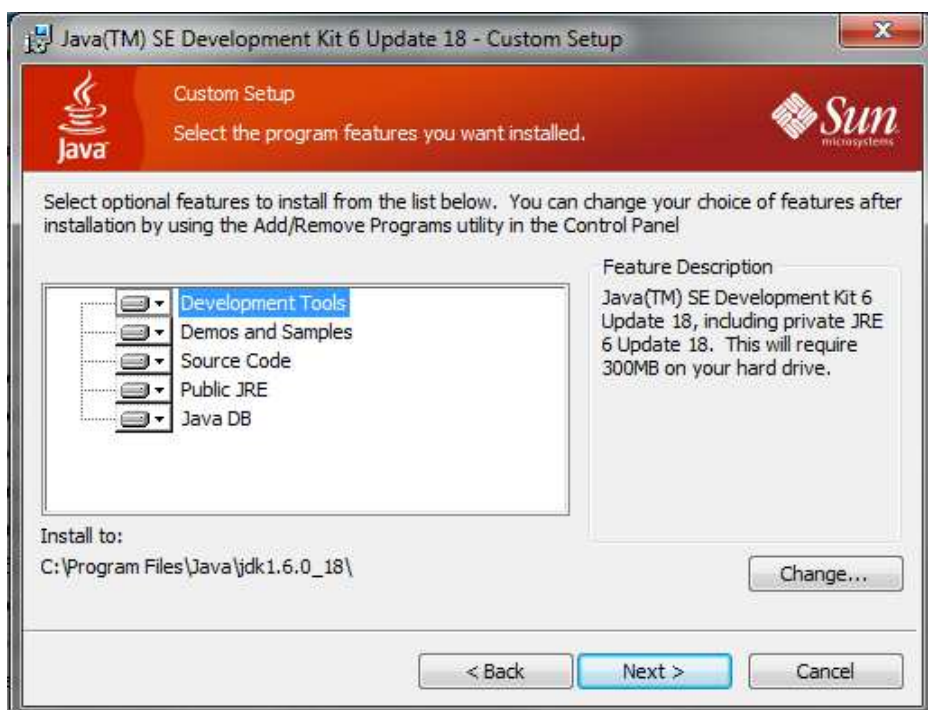
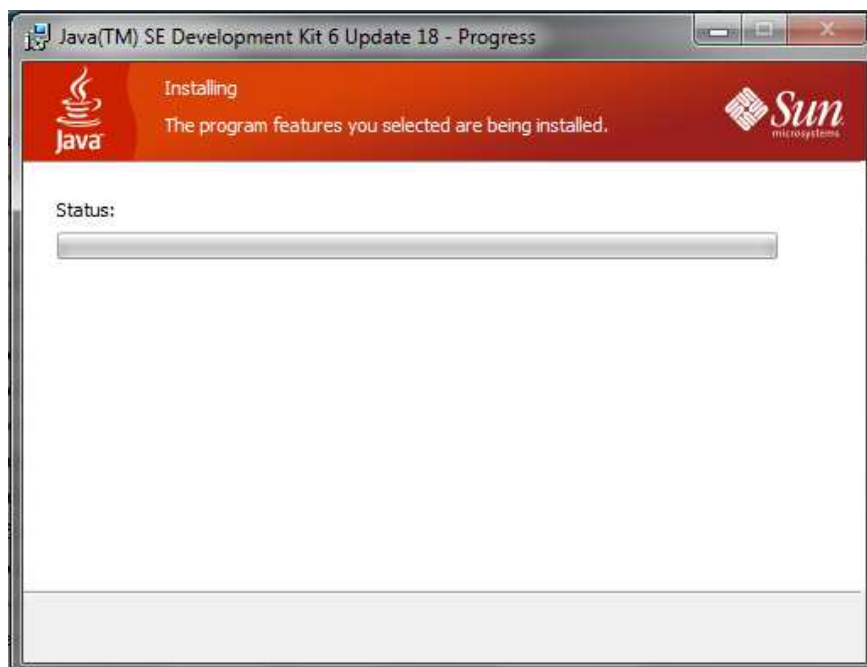


Figura 96. Asistente de instalación de Java Development kit



Si se pulsa en Change... se puede seleccionar la ruta donde se desea que se lleve a cabo la instalación. Si no se indica, se instalará en la ubicación por defecto. A continuación se ha de pulsar Next>. La instalación de Java JDK empezará y serán necesarios unos minutos para que se complete.



*Figura 97 Asistente de instalación de Java Development kit*

Al terminar la instalación de Java JDK, se nos pedirá la carpeta de destino donde se tiene que instalar JRE de Java, que es un conjunto de utilidades que permite la ejecución de programas Java y por tanto, es importante que esté instalado en la máquina. A continuación es necesario pulsar en Next> para que comience la instalación.

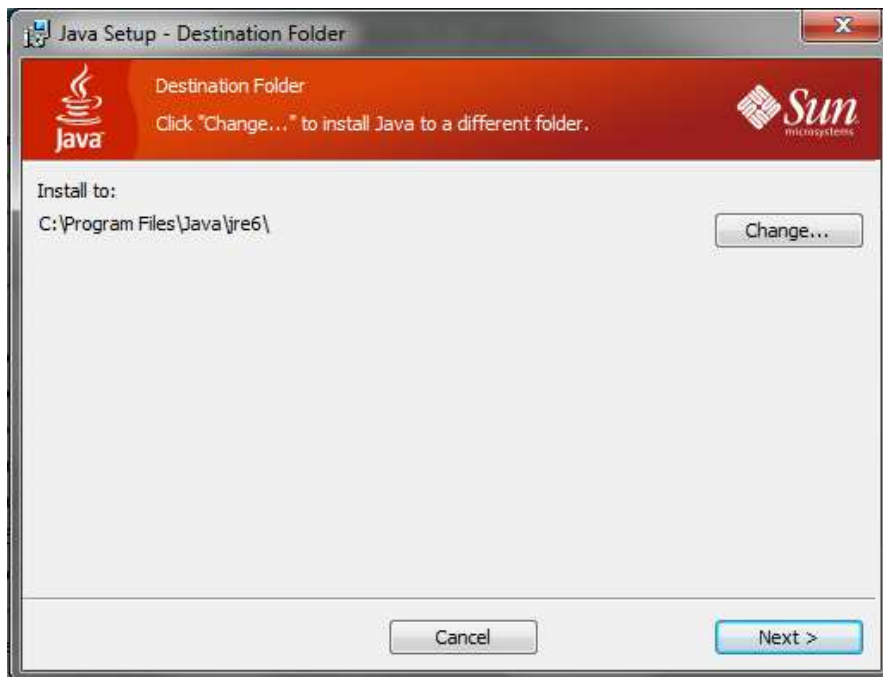
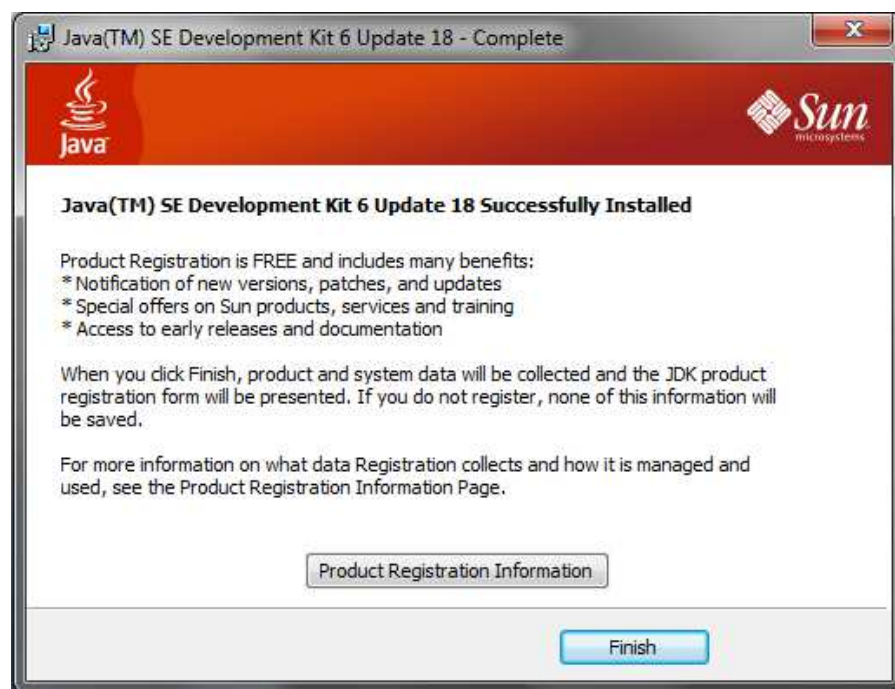


Figura 98 Asistente de instalación de Java Development kit



Figura 99 Asistente de instalación de Java Development kit

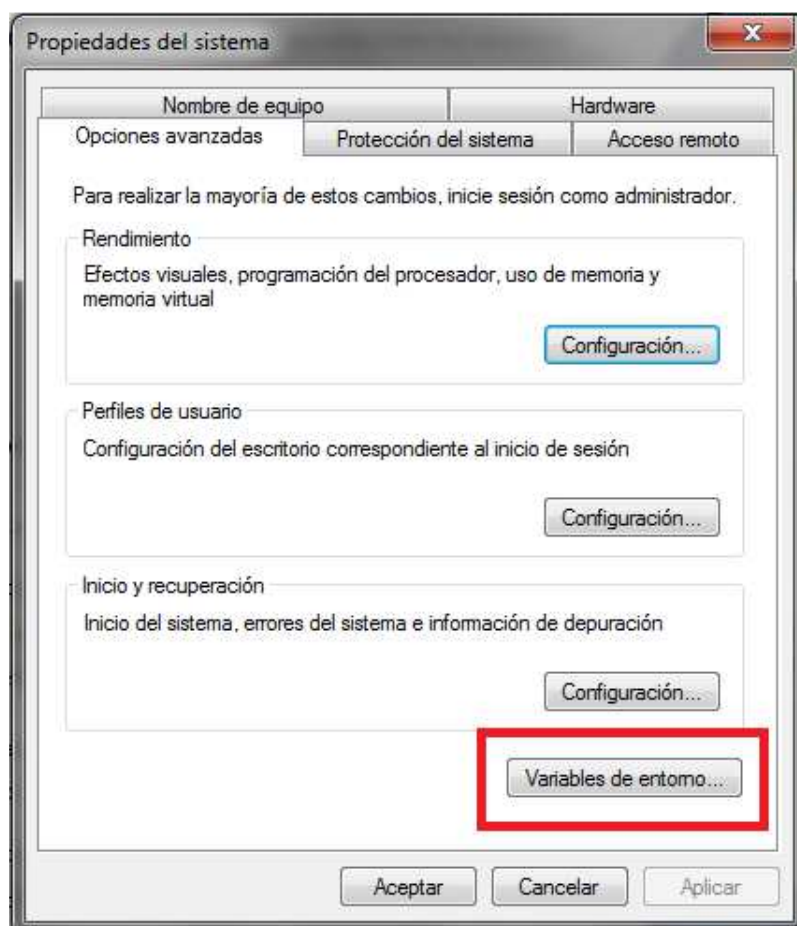
Una vez que haya terminado la instalación de Java JRE se mostrará una pantalla como la que se puede ver en la Figura 100. Al pulsar en Finish terminará la instalación de Java JDK y Java JRE.



*Figura 100 Asistente de instalación de Java Development kit*

Una vez finalizada la instalación, es necesario configurar las variables de entorno del equipo para poder compilar el código desde la Consola de Comandos de Windows 7.

Para ello, se ha de pulsar clic derecho a **Mi PC** y seleccionar **Propiedades**. Aparecerá una ventana que contiene la información de nuestro sistema. En esta ventana se ha de hacer clic en **Configuración avanzada del sistema**. Deberá de aparecer una ventana como la mostrada en la Figura 101.



*Figura 101 Ventana de propiedades del sistema*

En la pestaña **Opciones Avanzadas** se ha de hacer clic en **Variables de entorno**. Aparecerá una ventana como la mostrada en la Figura 102.

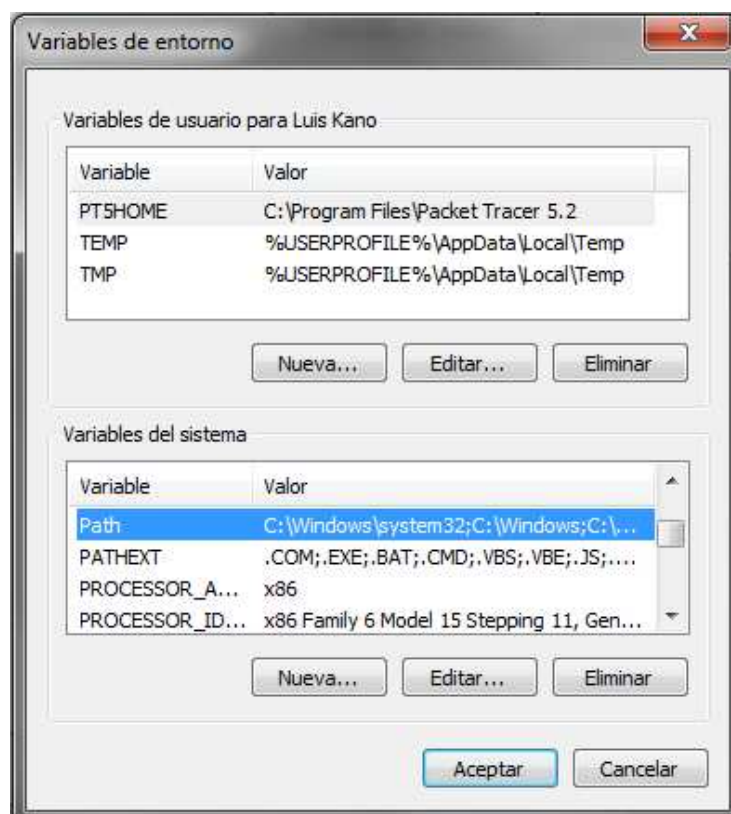


Figura 102 Ventana variables de entorno

En las **Variables del sistema**, se ha de seleccionar la variable llamada **Path** y pulsar en **Editar**. Se abrirá una nueva ventana como la que se puede ver en la Figura 103 y se ha de agregar lo siguiente **;C:\Program Files\VERSION DE JAVA\bin** en caso de que se seleccionara la ruta por defecto en la instalación. En caso contrario, se ha de poner la ruta elegida en la instalación de JDK. Una vez añadida la ruta, se ha de pulsar en aceptar.

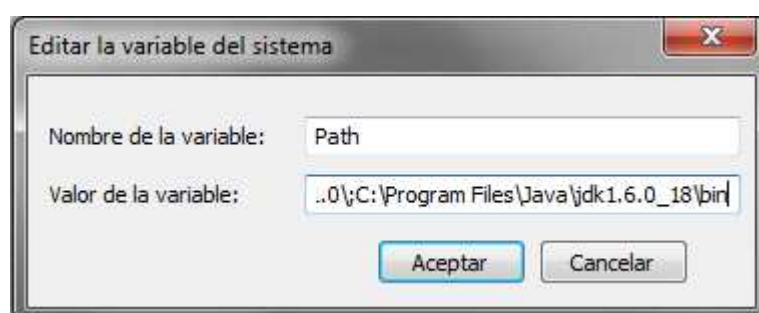


Figura 103 Ventana para editar las variables de entorno del sistema

A continuación, y en la ventana de **Variables del sistema** se ha de crear una nueva variable llamada **Classpath** que tendrá la dirección del src.zip. En el caso de haber seleccionado la ruta por defecto en la instalación de JDK, la dirección será: **;C:\Program Files\Java\VERSION DE JAVA\src.zip**

A continuación se pulsa en aceptar.

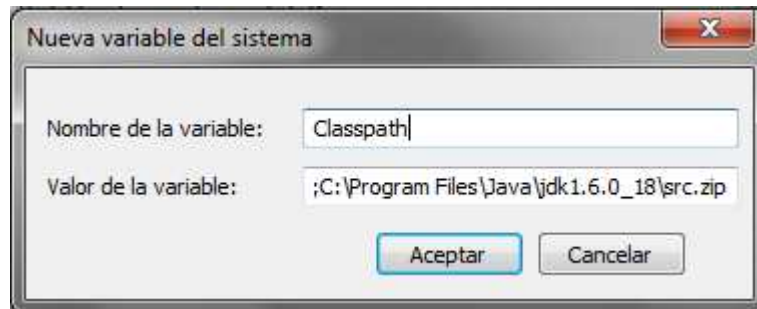


Figura 104 Ventana para añadir nueva variable de entorno al sistema

Se pulsa en aceptar en **Variables de Entorno** y en la ventana de **Opciones Avanzadas**.

Para comprobar que el proceso de instalación de JDK y JRE así como la configuración de las variables de entorno se ha realizado con éxito se puede ejecutar en la Consola de Comandos de Windows 7 el comando **javac**. Si el proceso ha finalizado correctamente, se debería de mostrar el contenido que se puede ver en la Figura 105.

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Todos los derechos reservados.

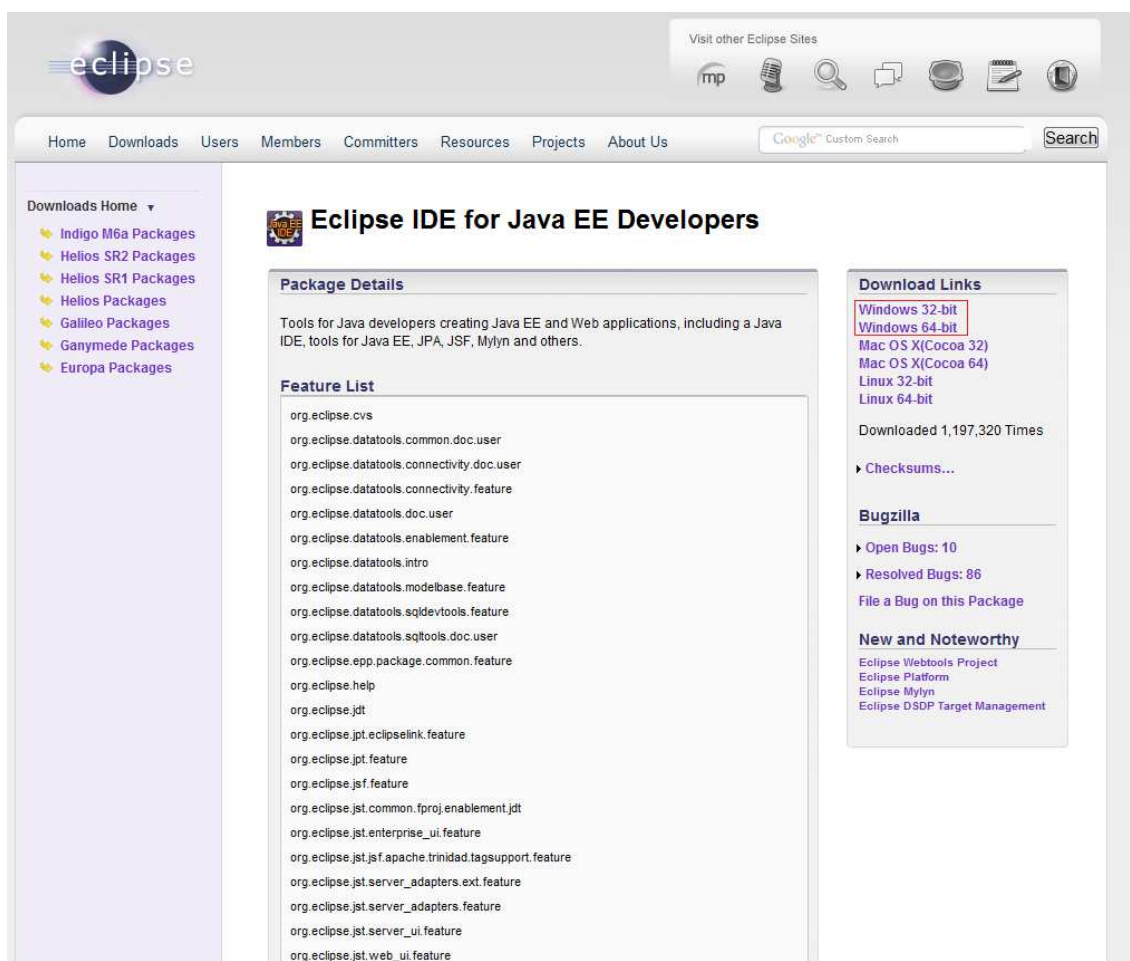
C:\Users\Luis Kano>javac
Usage: javac <options> <source files>
where possible options include:
  -g               Generate all debugging information
  -g:none          Generate no debugging information
  -g:<lines,vars,source> Generate only the specified debugging information
  -nowarn          Generate no warnings
  -verbose         Output message sequence numbers
  -deprecation     Output source location for each deprecation
  -classpath <path> Specify where to find the class files
  -cp <path>       Specify where to find the class files
  -sourcepath <path> Specify where to find the source files
  -bootclasspath <path> Override location of the bootstrap class files
  -extdirs <dirs>   Override location of the extension class files
  -endorseddirs <dirs> Override location of the endorsed class files
  -proc:<none,only> Control whether annotation processing is done.
  -processor <class1>[,<class2>,<class3>...]
                    Specify the annotation processor to run; bypasses default discovery process
```

Figura 105 Consola de comandos de Windows 7

## 2. Instalación de Eclipse

Eclipse es un entorno de desarrollo integrado para el desarrollo de aplicaciones. Para usar Eclipse es necesario, en primer lugar, descargar el paquete de Eclipse. [30]

Es importante que la versión de Eclipse que se descargue sea consistente con la versión de Java instalada en el ordenador.



*Figura 106 Descarga de Eclipse*

Una vez descargado el paquete, es necesario descomprimirlo en una carpeta situada en la ubicación que se desee ya que Eclipse no dispone de instalador, por lo cual para usar la herramienta solo hay que pulsar sobre el ejecutable incluido en los archivos del paquete.



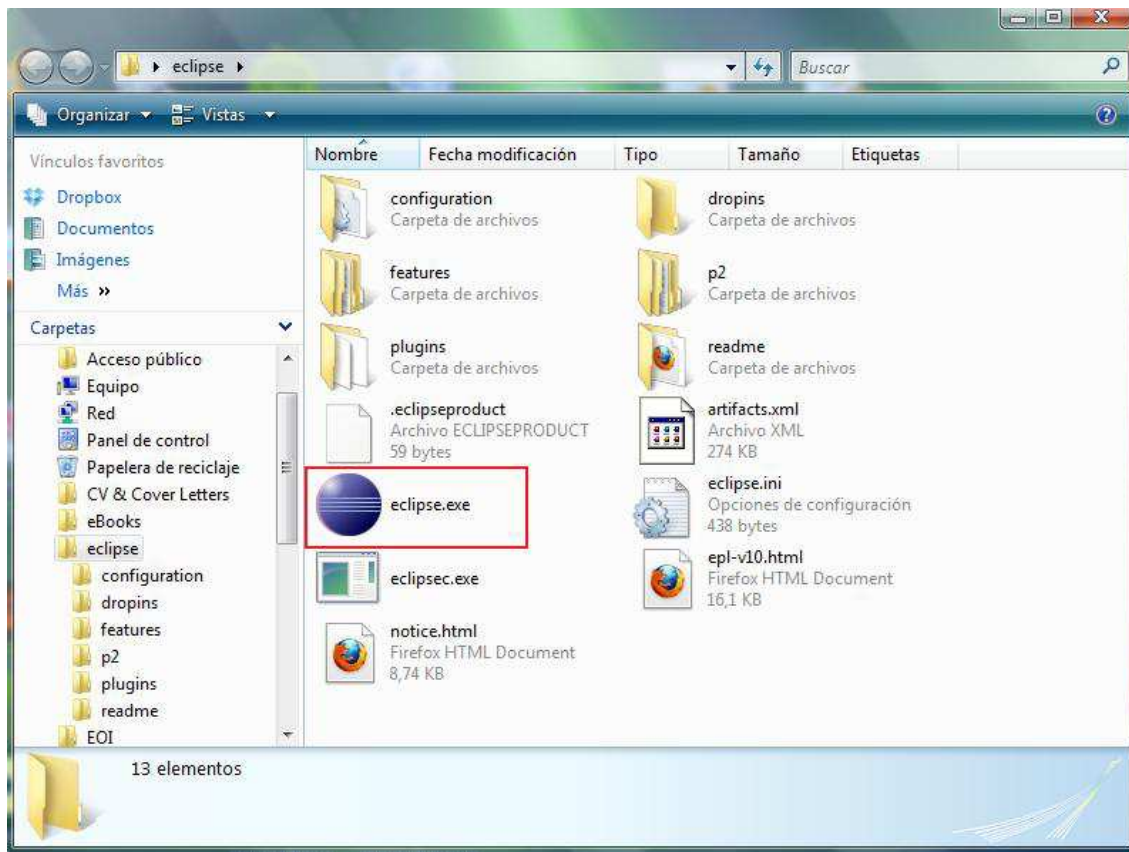


Figura 107 Ejecutable de Eclipse

### 3. Uso de la herramienta Eclipse

Quando se inicia la herramienta Eclipse, se solicita indicar el directorio en el que está guardado el proyecto que se va a abrir o en el que se desea crear un nuevo proyecto (*workspace*) tal y como se puede observar en la Figura 108.

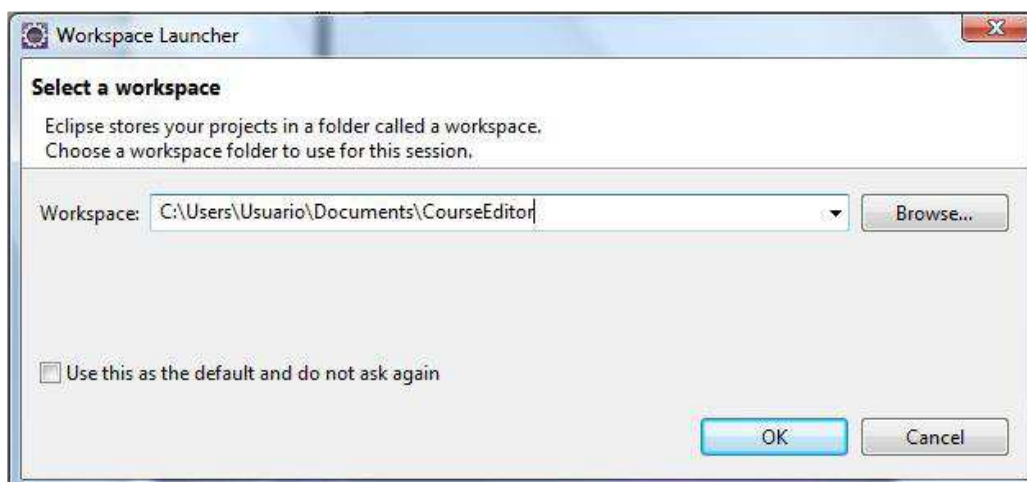
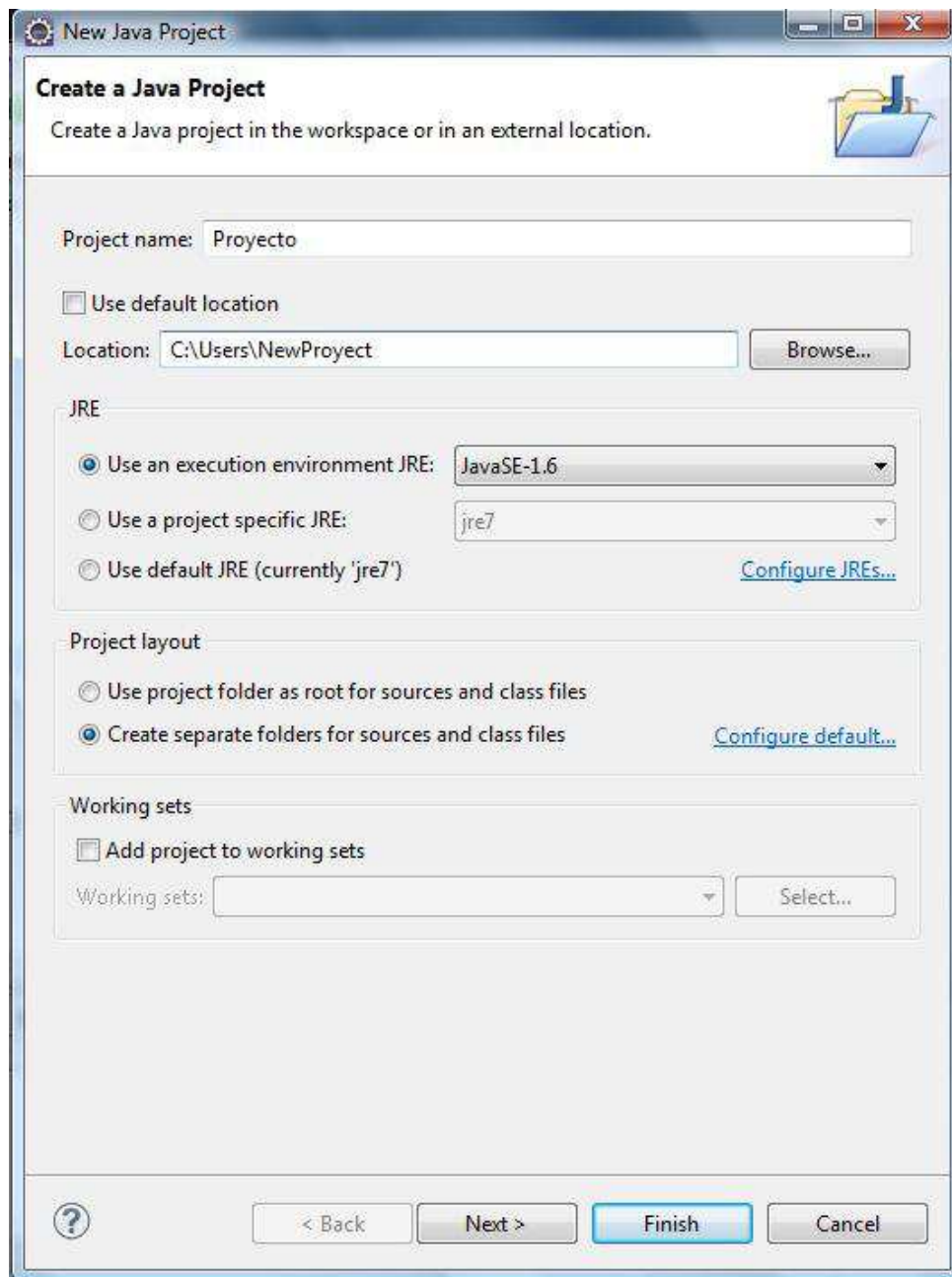


Figura 108 Selección de workspace en Eclipse



En el entorno de desarrollo Eclipse todo archivo se almacena dentro de un proyecto. Esto quiere decir que todo documento, carpeta, archivo de código fuente (.java) y código compilado (.class) tiene que estar contenido dentro de un proyecto.

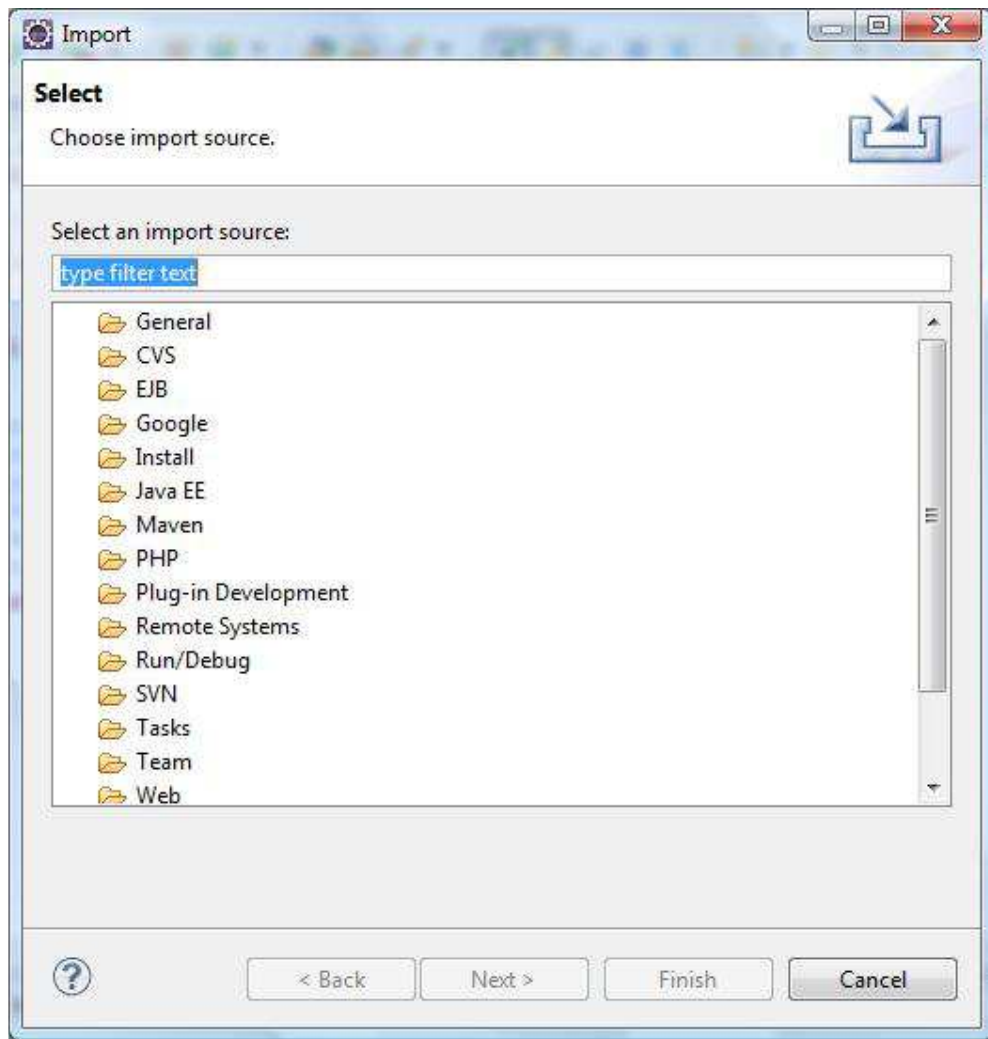
Para programar en Java, se ha de crear un nuevo "Java Project" seleccionando "**Java >Java Project**" rellenando el nombre que tendrá el proyecto y haciendo clic en el botón "**Next >**" del asistente de creación.



*Figura 109 Asistente de creación de un nuevo proyecto*

Para editar un elemento existente debe de ser primero importado dentro de un proyecto de Eclipse. Esto puede hacerse desde el menú "**File > Import...**" o bien pulsando con el botón derecho en cualquier punto de la vista del "*Package Explorer*" o

del "*Resource Navigator*". Del mismo modo, en Eclipse pueden importarse proyectos ya existentes siguiendo el procedimiento mencionado anteriormente.

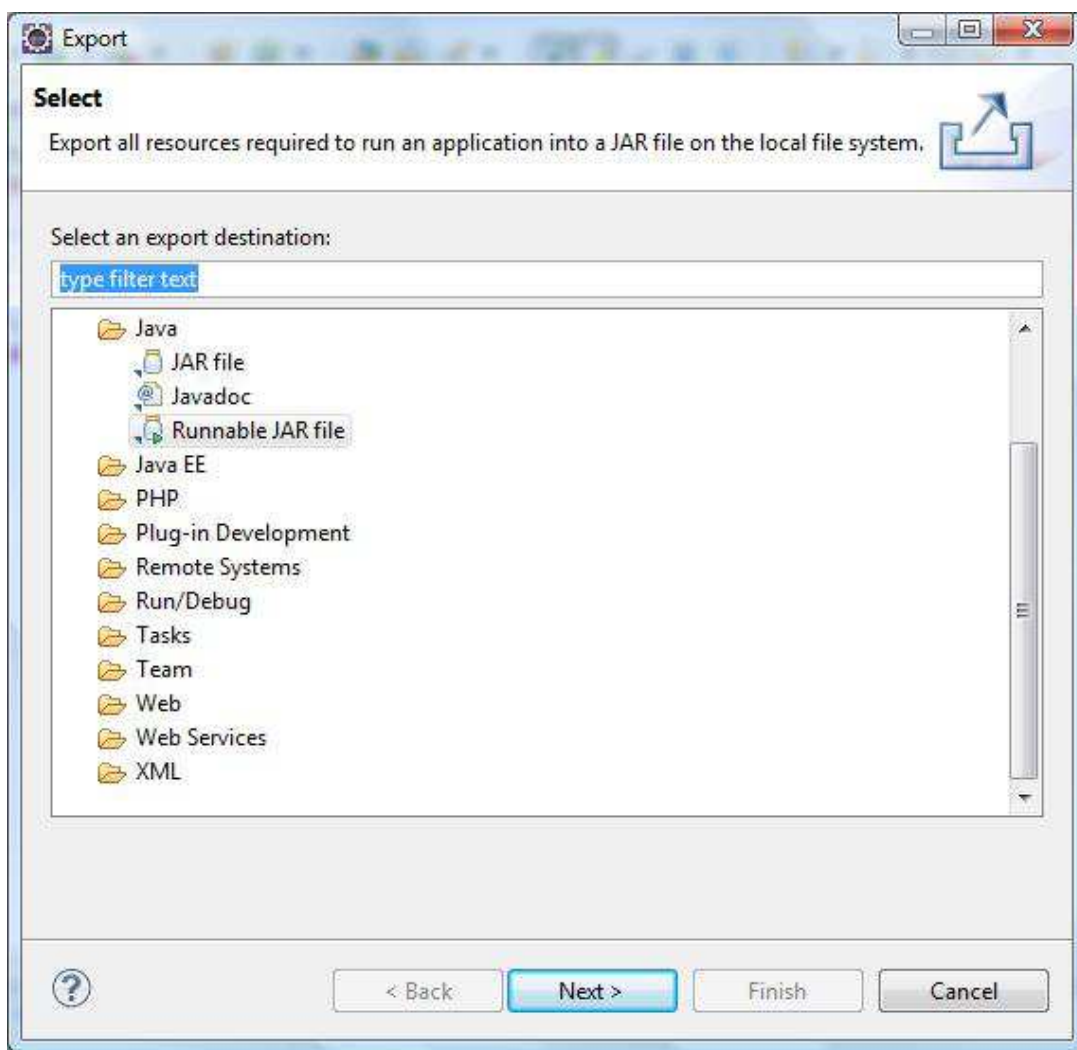


*Figura 110 Ventana para importar proyectos existentes a Eclipse*

Es importante tener en cuenta que todo elemento que se importa en Eclipse se duplica. Esto significa que borrar la copia que Eclipse esté manejando no borrará el archivo original, de manera que se seguirá teniendo una copia de reserva.

Sin embargo, si se usa la opción de importar un proyecto de Eclipse ya existente, los contenidos de dicho proyecto serán duplicados. Así que hay que ser especialmente cuidadoso al borrar proyectos importados del entorno de trabajo de Eclipse, ya que es posible que otras copias de backup de dicho proyecto no existan.

Todo archivo creado mediante la utilización de Eclipse puede ser exportado como un archivo normal (seleccionando **Export...** > **File System**), como un archivo ".jar" e incluso como archivos comprimidos en ".zip". El proceso que hay que seguir es similar al proceso recientemente explicado para importar archivos, sólo que seleccionando ahora la opción "**Export**".



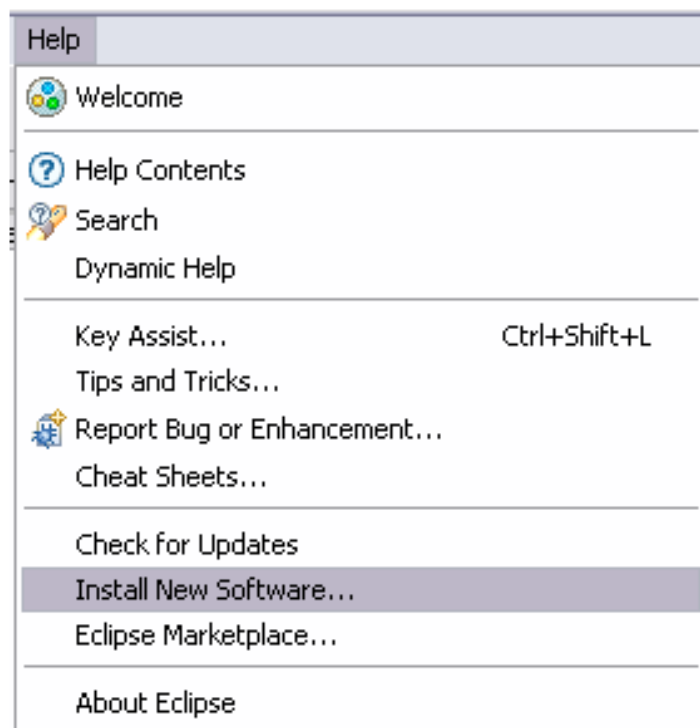
*Figura 111 Ventana para exportar archivos desde Eclipse*

#### 4. Código de la aplicación

El código de la herramienta CourseEditor se encuentra en un repositorio alojado en la URL: <https://svn.gast.it.uc3m.es/PFCs/CourseEditor/CourseEditor>

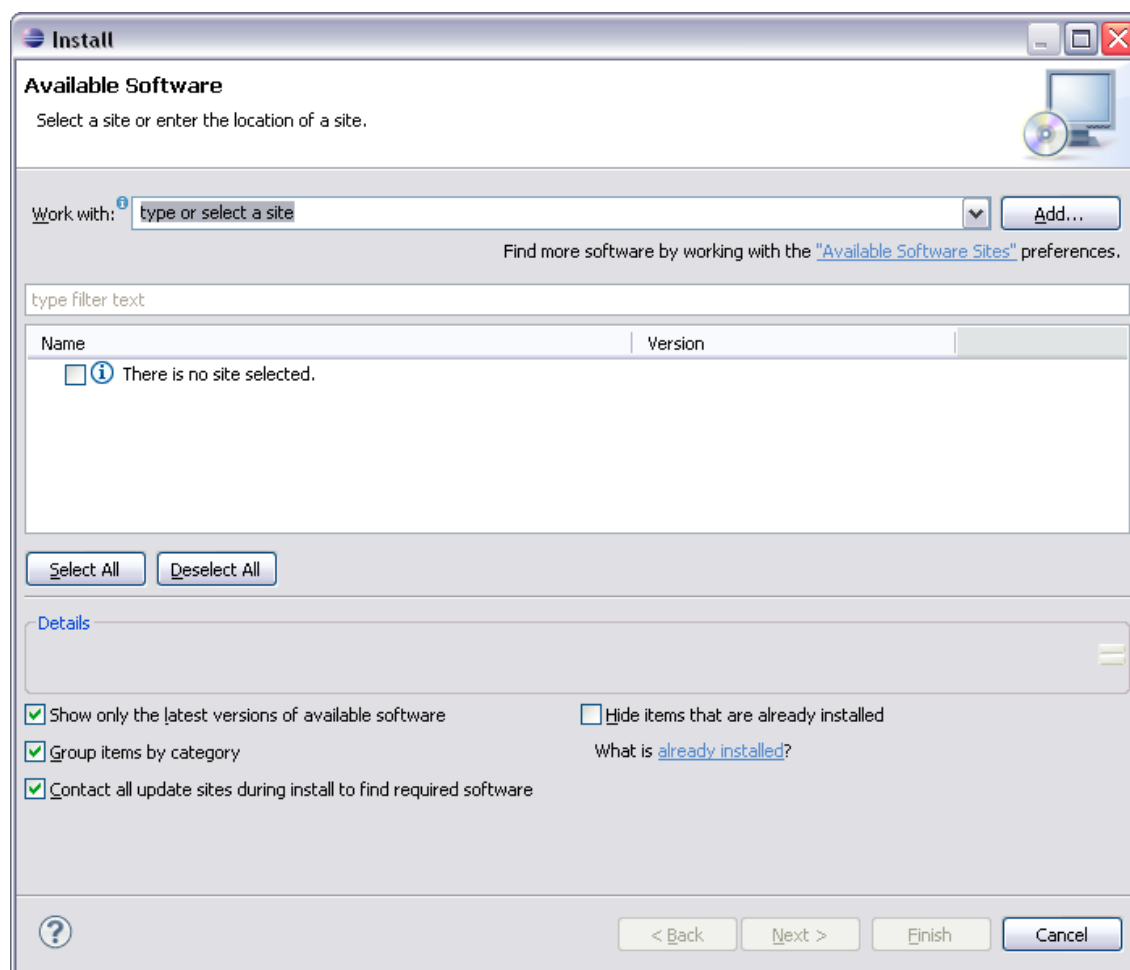
Para poder obtener el código de la aplicación desde el repositorio es necesario disponer del plugin Subversion instalado en Eclipse. A continuación se mostrará como instalar dicho plugin.

En el menú, se ha de seleccionar “**Help > Instal New Software...**”



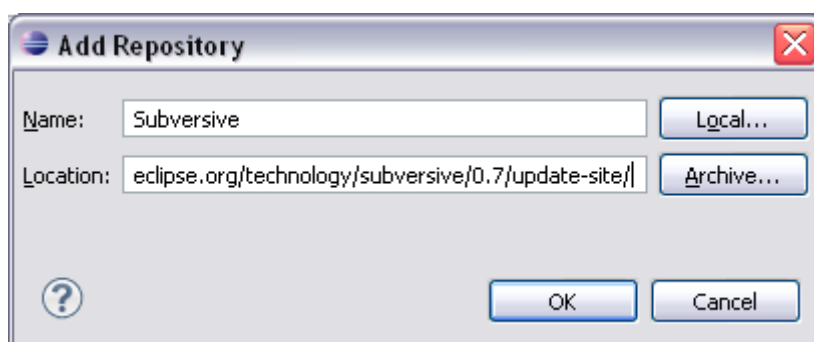
*Figura 112 Opciones de la pestaña 'Help' del menú de Eclipse*

A continuación se ha de seleccionar el repositorio donde se encuentra el software, en este caso *Subversive* y pulsar en el botón 'Add'



*Figura 113 Asistente de instalación de nuevo software en Eclipse*

Una vez seleccionado el repositorio, se ha de indicar tal y como se puede ver en la Figura 114, la URL en la que se encuentra la actualización de Subversión. Esta URL puede ser consultada en el sitio web de Eclipse (<http://www.eclipse.org/subversive/downloads.php>.)



*Figura 114 Añadir repositorio*

A continuación se ha de seleccionar las características de Subversion que se desean instalar en la herramienta Eclipse y pulsar 'Next'

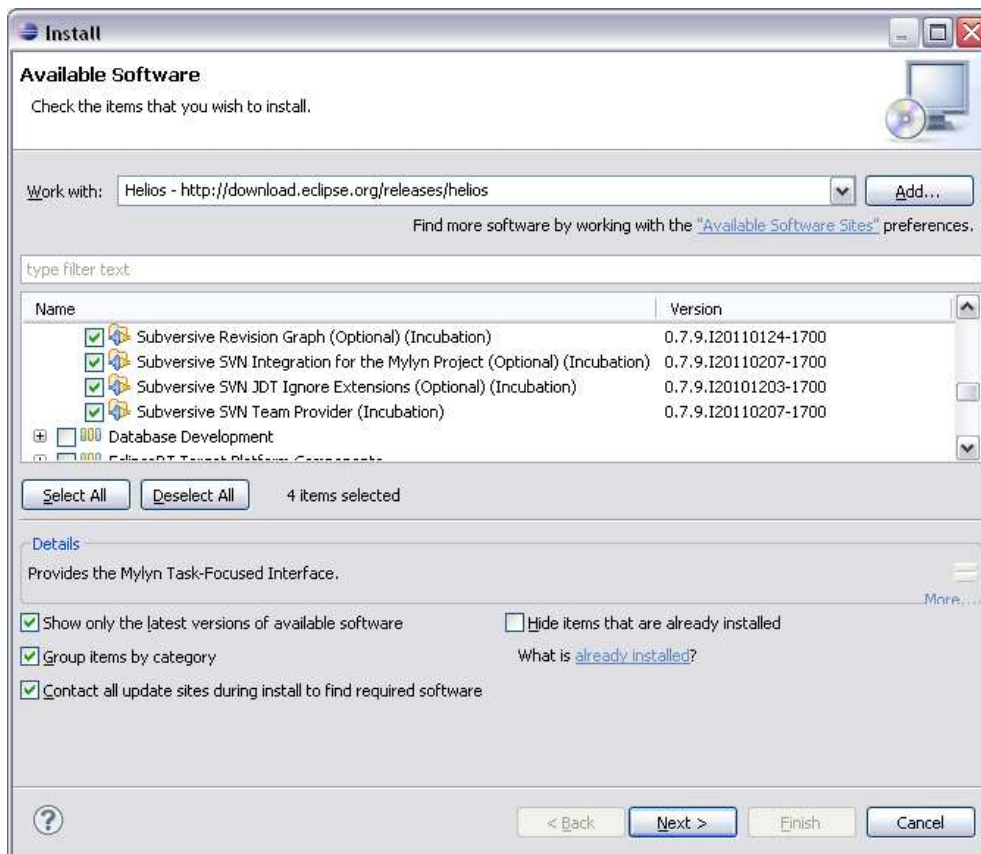


Figura 115 Asistente de instalación de nuevo software en Eclipse

El asistente de instalación muestra un listado de las características a instalar tal y como se puede observar en la Figura 116.

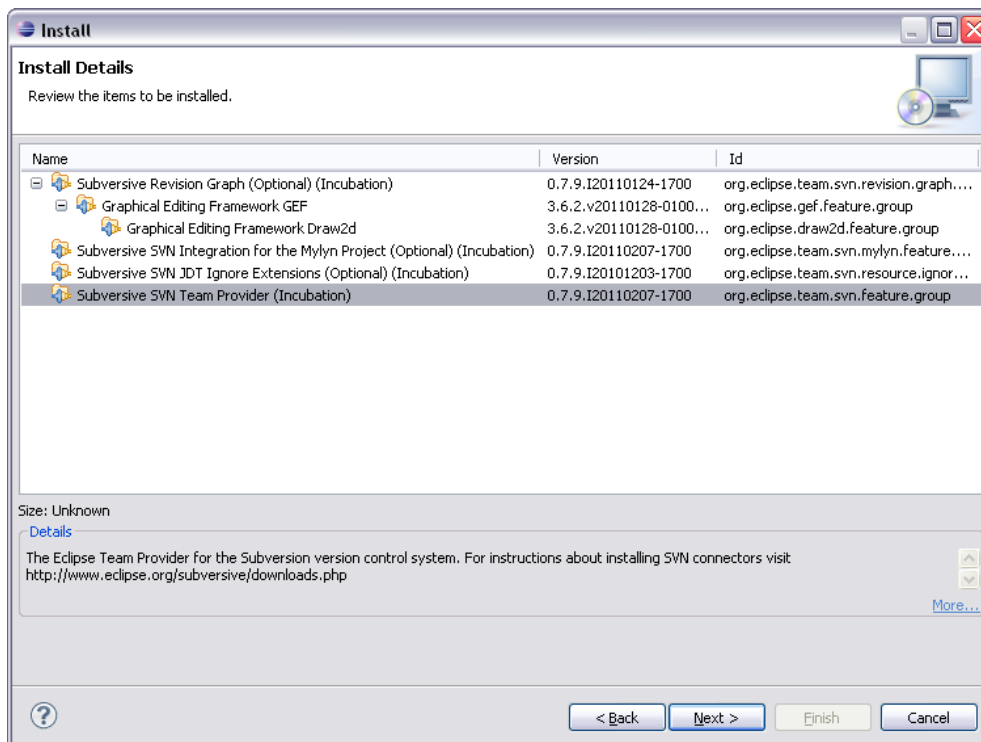
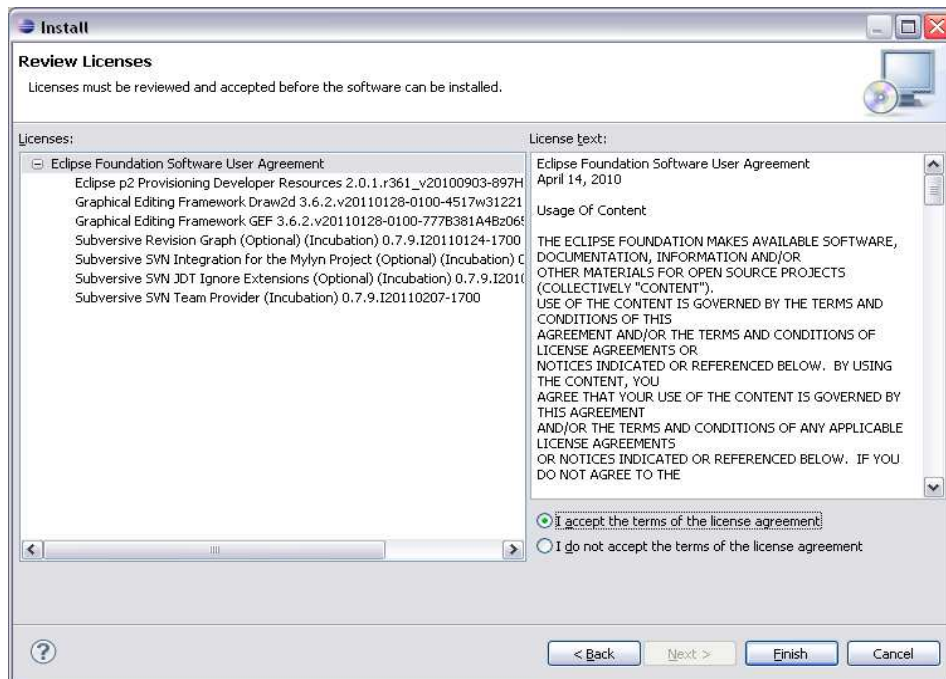


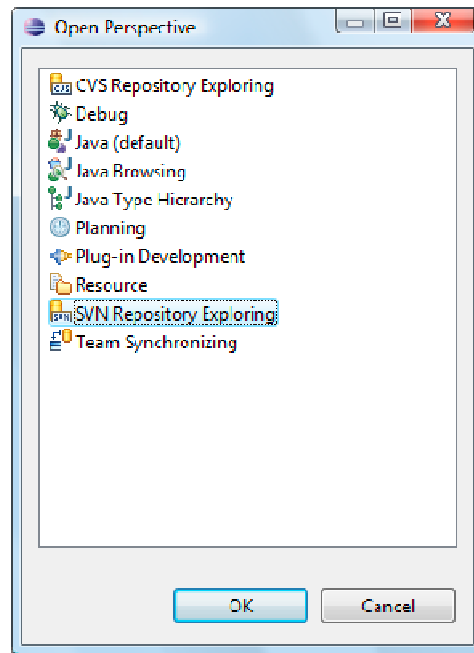
Figura 116 Asistente de instalación de nuevo software en Eclipse

Por último, para dar comienzo a la instalación se han de aceptar los términos de licencia y reiniciar Eclipse una vez instalado el plugin para que los cambios tengan efecto.



*Figura 117 Asistente de instalación de nuevo software en Eclipse*

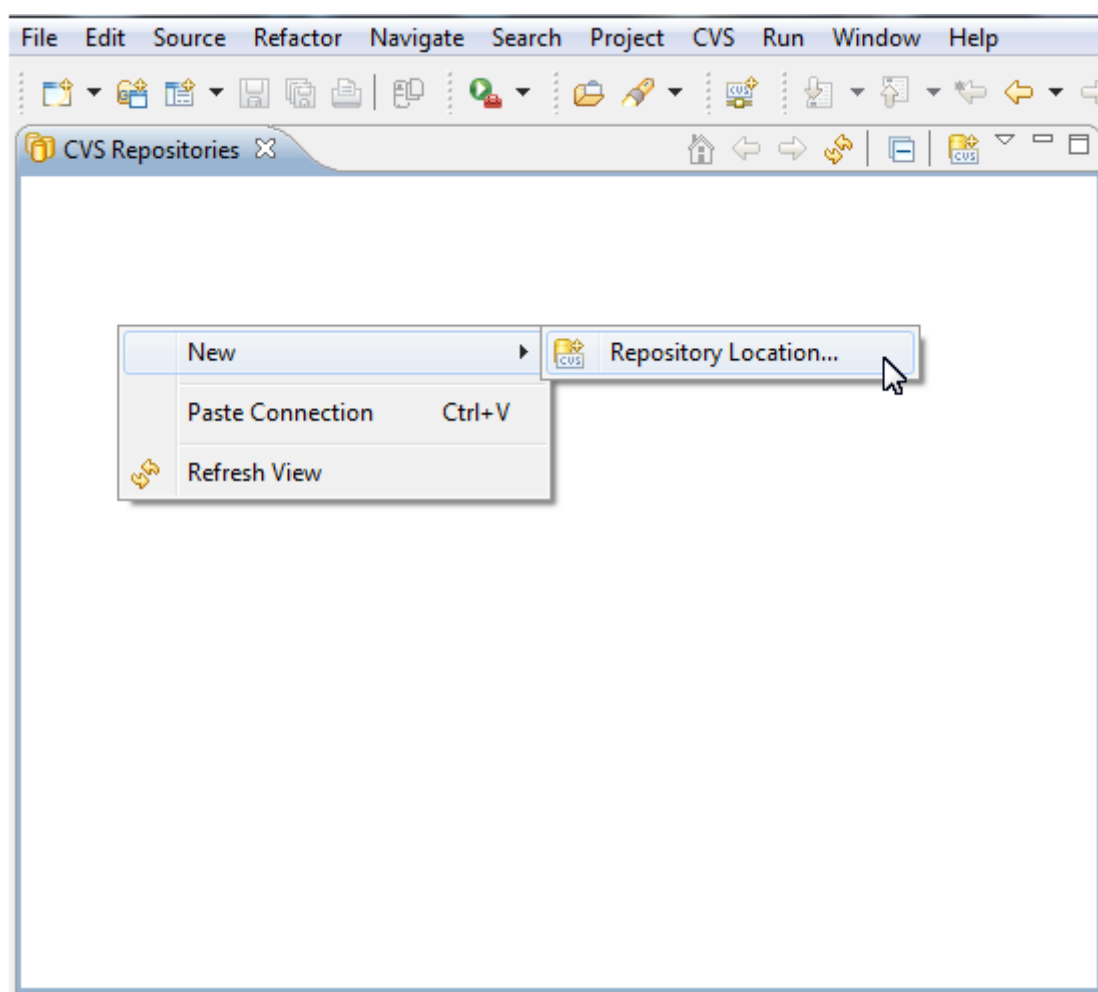
Una vez instalado el plugin, se está en disposición de acceder a la perspectiva Subversive seleccionando '**Window > Open Perspective > Other...**'



*Figura 118 Ventana para abrir distintas perspectivas en Eclipse*

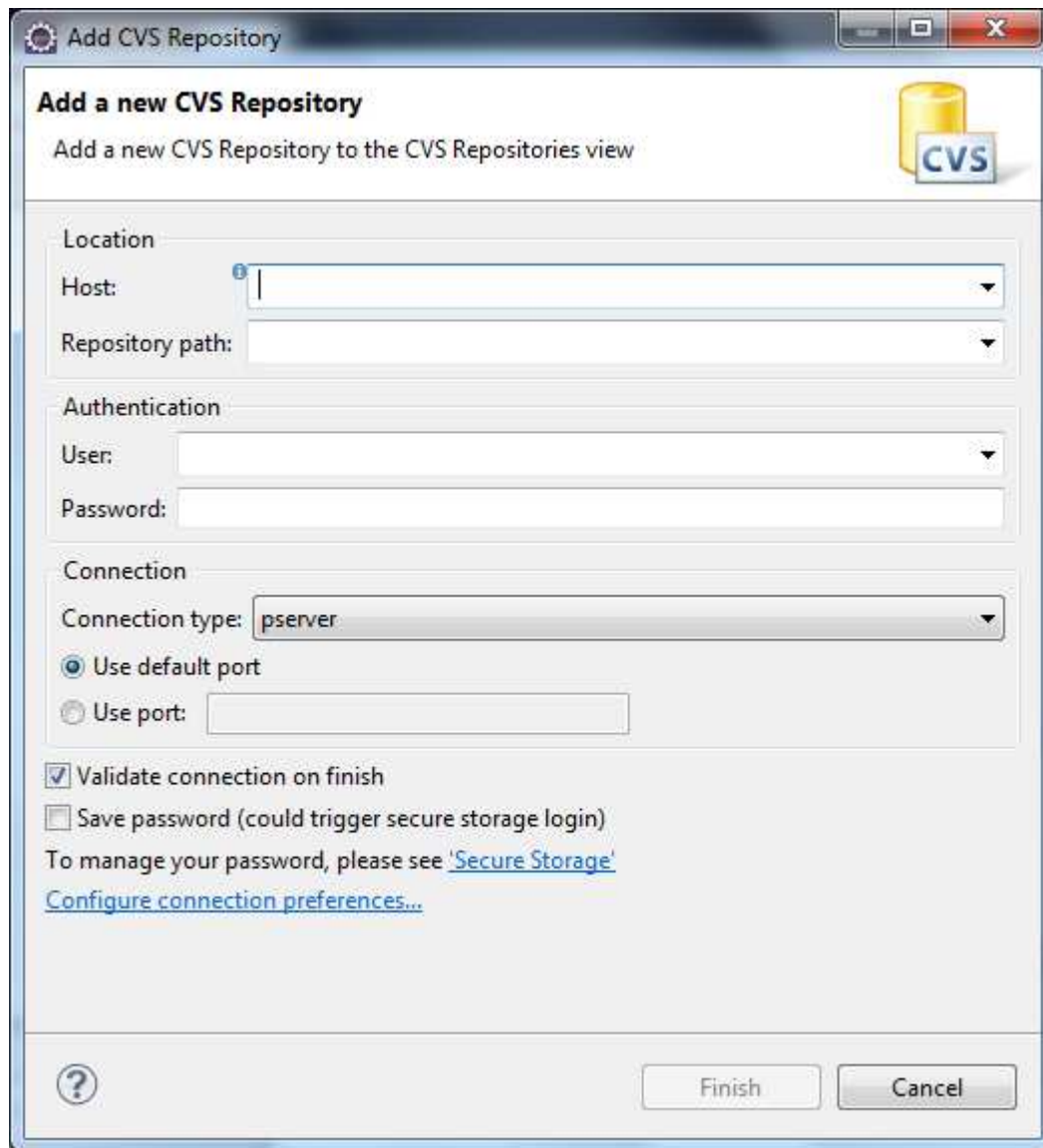
Para poder descargar a local el código de la herramienta CourseEditor, desde la perspectiva de Subversive se ha de crear un nuevo repositorio. Para ello se selecciona la opción “**New>Repository Location...**”





*Figura 119 Creación de un nuevo repositorio*

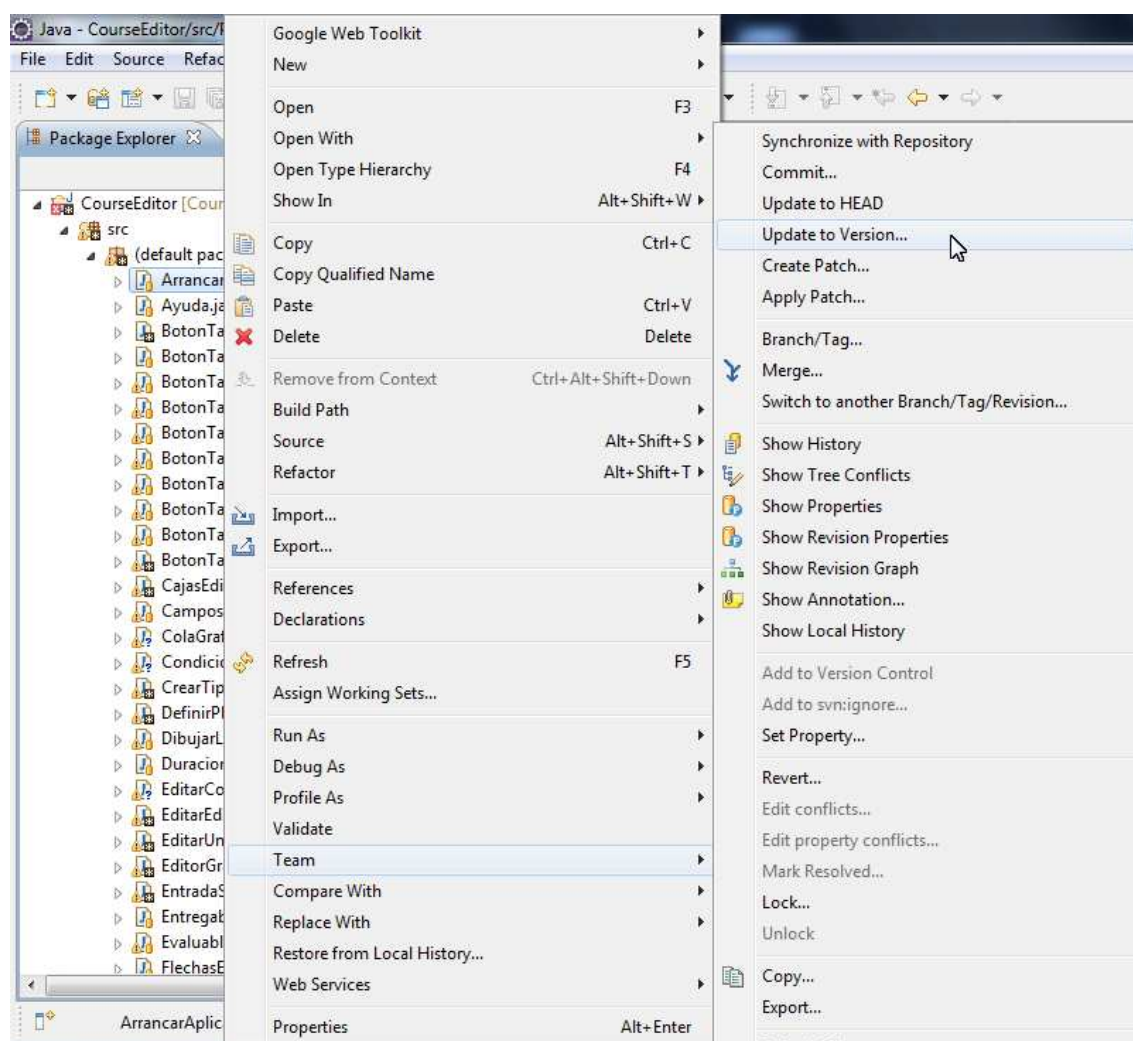
A continuación en el formulario que se puede observar en la Figura 120 se ha de introducir los datos necesarios tanto del host y ubicación del repositorio (<https://svn.gast.it.uc3m.es/PFCs/CourseEditor/CourseEditor>) como los datos de autenticación proporcionados por los administradores de forma que se pueda crear el nuevo repositorio.



*Figura 120 Añadir nuevo repositorio*

Una vez configurado el repositorio, es importante destacar dos operaciones básicas mediante las cuales podremos descargar el código de la herramienta CourseEditor (Update) o subir una nueva versión del mismo al repositorio (Commit).

Mediante la opción Update es posible descargar la última versión del proyecto o de una clase en particular así como descargar una versión anterior si fuese necesario. Para ello sobre la clase (o la carpeta del proyecto si fuera el caso) se ha de seleccionar **“Team>Update to HEAD...”** o **“Team>Update to version...”** y seleccionar la versión que se desea descargar.



*Figura 121 Selección de la opción 'Update to version'*

Mediante la opción 'Commit' es posible subir al repositorio una nueva versión de una o varias clases del proyecto o incluso de la totalidad del mismo. Para ello sobre la clase (o la carpeta del proyecto si fuera el caso) se ha de seleccionar **"Team>Commit..."**. Es posible añadir un comentario acerca de la nueva versión que se sube al repositorio para, por ejemplo, indicar los cambios introducidos o el número de versión.

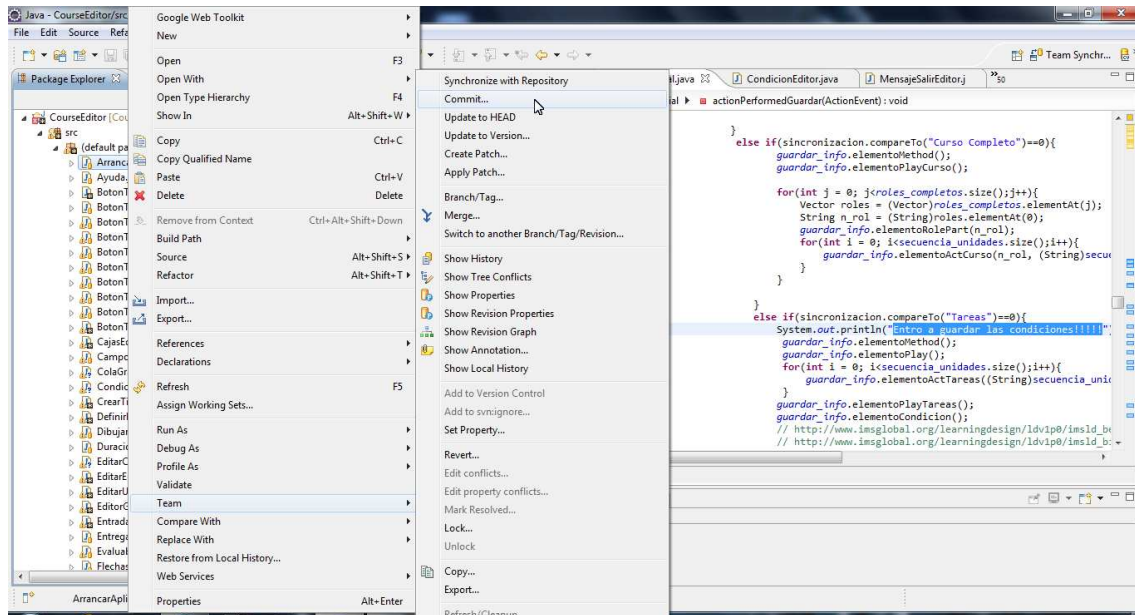


Figura 122 Selección de la opción Commit

### 5. Principales clases y métodos de la aplicación

En este apartado se realizará una explicación acerca de la estructura del proyecto CourseEditor y de cada una de las clases que lo forman con el objetivo de proporcionar al usuario una visión general del proyecto y que por tanto sea más fácil realizar modificaciones sobre el mismo.

La estructura de carpetas del proyecto Eclipse de la aplicación es la siguiente:

- Src

Esta carpeta contiene las el código de la aplicación. Está formada por dos paquetes:

- Default package. Contiene las clases java de la aplicación.
- Images. Contiene las imágenes de la aplicación.

- Ejecutables

En esta carpeta se almacenan los ejecutables creados a partir del código de la aplicación.

- Help

En esta carpeta se encuentran los ficheros necesarios para crear la ayuda de la aplicación.

- Lib

Aquí se encuentran las librerías adicionales que se deseen usar en la aplicación.

A continuación se presentan las clases que forman la aplicación junto con una breve descripción de la funcionalidad que implementa cada una.

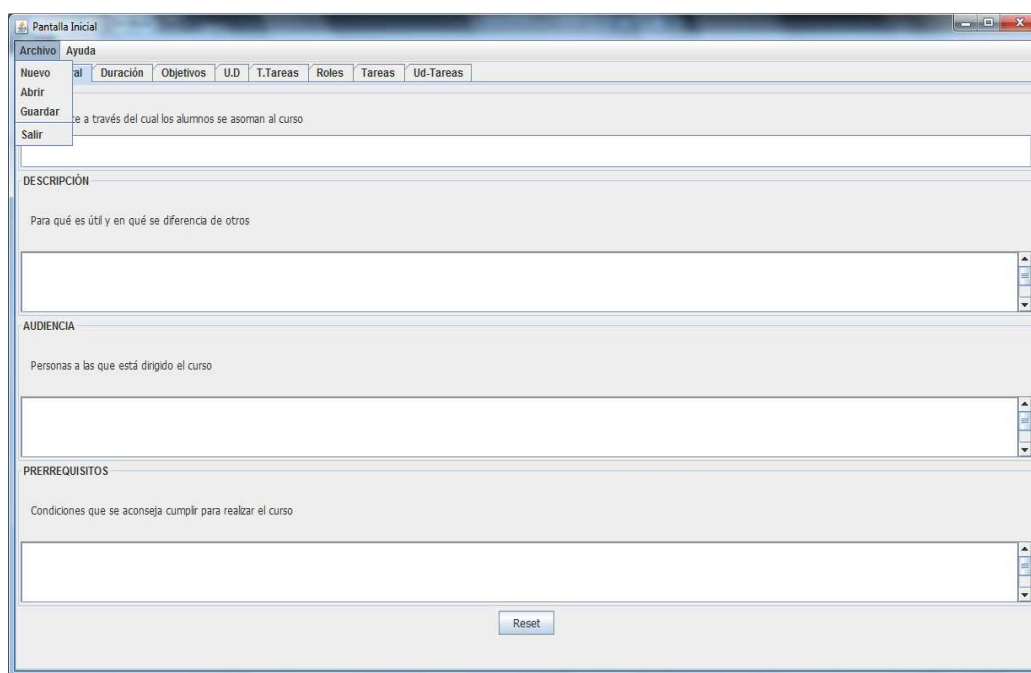
- **ArrancarAplicacion**

Es la clase en la que se encuentra el método main, y por lo tanto es la que se ejecutará para lanzar la aplicación. El método main consiste en una llamada al método PantallaInicial que se describe a continuación.

- **PantallaInicial**

Ésta es la clase principal de la aplicación. Es la encargada de la interfaz principal con la que el usuario trabaja. Esta pantalla consta de ocho pestañas (Info. General, Duración, Objetivos, U.D, T.Tareas, Roles, Tareas y Ud-Tareas) que el usuario tiene que rellenar para completar la definición de un curso completo.

A su vez, en la parte superior también hay definido un menú desde el cual el usuario puede realizar las principales opciones tales como abrir y cargar un curso previamente creado, guardar el que se está definiendo, salir de la aplicación o visualizar la ayuda de la herramienta.



*Figura 123 Pantalla Inicial de CourseEditor*

La estructura de esta clase es la siguiente y sus principales métodos son los siguientes:

## CAPÍTULO: Tutorial para la realización de modificaciones en la herramienta.

- En la parte superior de la clase se definen todos los atributos tales como paneles, etiquetas, botones etc. que posteriormente se utilizarán.
- El constructor **public** `PantallaInicial(int h, int v, boolean visible)`, el cual hace visible la ventana a mostrar y llama al método `initComponents()`.
- El método `initComponents()` es el encargado de inicializar todos los componentes de los que consta la pantalla inicial.
- El método `private void actionPerformedContinuarDespues (java.awt.event.ActionEvent evt)` hace una llamada a la clase `MensajeSalir`, de forma que aparece una pequeña ventana que pregunta al usuario si desea guardar la información introducida antes de salir de la aplicación.
- Por cada pestaña se definen dos métodos:  
`private void actionPerformedX (javax.swing.event.ChangeEvent changeEvent)`  
y `private void actionPerformedX (java.awt.event.ActionEvent evt)`, donde X es un número que corresponderá con cada una de las pestañas.

De cada uno de estos métodos, el que recibe como parámetro un objeto *changeEvent*, se ejecutará cada vez que el usuario cambie de pestaña, ya que es el encargado de guardar de forma automática la información que el usuario ha introducido, sin necesidad de pulsar a ningún botón. Por otra parte, el que recibe como parámetro un objeto *ActionEvent* se ejecutará cada vez que el usuario pulse alguno de los botones de las pestañas, como el botón *Añadir* o de *Reset*.

- El método `private void actionPerformedGuardar (java.awt.event.ActionEvent evt)` es el encargado de guardar toda la información y llamar a los métodos que se encargaran de la escritura de los ficheros que se guardarán en el zip, tanto el documento de texto que permite cargar el curso posteriormente en la herramienta, como todos los archivos XML que se crean (el que sigue la especificación IMS-LD y los complementarios).
- El método `public void elegir()` es llamado cuando un usuario desea cargar un curso previamente guardado. Es el encargado de que aparezca el dialogo que permite al usuario seleccionar el archivo que desea cargar. Guarda la ruta elegida, descomprime el zip, selecciona el fichero de texto que sirve para la carga de información y llama al método encargado de su lectura.
- El método `private void mostrarAyuda()` es el encargado de mostrar la ventana de ayuda de la aplicación
- Finalmente la clase cuenta con una serie de métodos *get* y *set* que permiten obtener o asignar los valores contenidos en tablas, calendarios, etc.

- **BotonTabla**  
**BotonTablaEditar**  
**BotonTablaRoles**  
**BotonTablaRolesEditar**

**BotonTablaTarea**  
**BotonTablaTareaEditar**  
**BotonTablaTipoTareas**  
**BotonTablaTipoTareasEditar**  
**BotonTablaUnidades**  
**BotonTablaUnidadesEditar**

Todas estas clases definen los botones *Editar* y *Borrar* de cada una de las tablas de la aplicación y su funcionalidad

- **CajasEditor**

Esta clase define los atributos que poseerá una caja del editor gráfico, de forma que nos permita unir distintos atributos en un mismo elemento. Los atributos definidos en esta clase son:

- *String identificador*: Identificador asignado a la caja en cuestión.
- *String nombre*: Nombre de la caja
- *String tipo*: Tipo de tarea que define la caja.
- *Vector roles*: Roles que intervienen en la tarea que define la caja
- *Vector padres*: El/los nodos padre/s de la caja en la secuenciación
- *Vector hijos*: El/los nodos hijo/s de la caja en la secuenciación
- *String pos\_X*: posición de la caja en el eje de las X
- *String pos\_Y*: posición de la caja en el eje de las Y
- *boolean visitado*: Boolean que indica si esa caja ha sido o no visitada para su reconstrucción.

La clase incluye dos constructores que utilizan distintos parámetros de los arriba mencionados, así como métodos *get* y *set* para cada uno de ellos.

- **Campos**

Al igual que la clase anterior, la clase Campos define los atributos que posee una tarea:

- *String Nombre*: nombre asignado al campo y
- *String Valor*: Valor que poseerá el campo.

Esta clase cuenta con dos constructores y los métodos *get* y *set* correspondientes.

- **ColaGrafo**

Este método permitirá añadir y obtener los nodos añadidos en el editor gráfico cuando el mismo se recupera desde el fichero XML guardado.

Para la reconstrucción de un grafo guardado, se ha implementado un sistema de colas, el cual permita recorrer todos y cada uno de los nodos del gráfico. La aplicación guardará la información de las cajas del editor. Esta información será leída y añadida al grafo en reconstrucción.

- **CondicionEditor**

Esta clase define los atributos que poseerá una condición del editor gráfico, de forma que nos permita unir distintos atributos en un mismo elemento. Los atributos definidos en esta clase son:

- *String unidad*: Unidad didáctica a la que pertenece la condición
- *String nombre*: nombre de la condición
- *String si*: condición que se debe de cumplir
- *String entonces*: tarea que se realizará si se cumple la condición
- *String sino*: tarea que se realizará si no se cumple la condición
- *int nota*: nota que el alumnos ha obtenido en una actividad evaluable
- *boolean entregable*: boolean que define si la actividad es entregable o no.
- *boolean evaluable*: boolean que define si la actividad es evaluable o no.

Esta clase cuenta con un constructor y los métodos *get* y *set* correspondientes.

- **CrearTipoTarea**

Esta clase define la ventana que le aparecerá al usuario cuando crea un nuevo tipo de tareas.

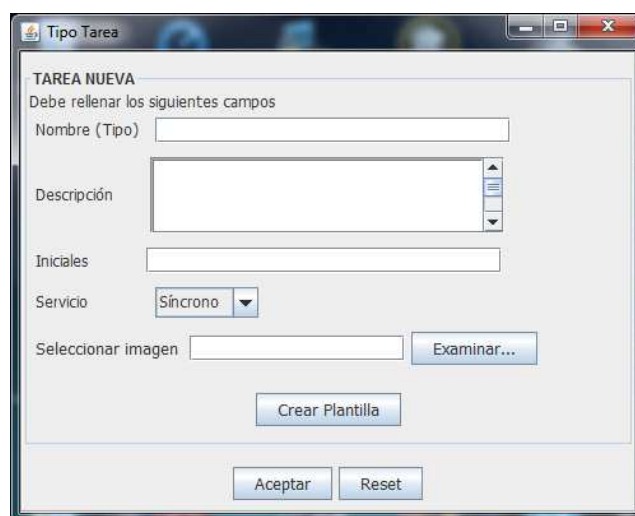


Figura 124 Ventana para crear nuevos tipos de tareas

Tiene la misma estructura que la clase *PantallaInicial*. Dicha estructura se explica a continuación.

- Definición de los atributos que se utilizarán
- Constructor que hace visible la ventana e invoca al método  *initComponents*
- Método *private void initComponents()* que inicializa todos y cada uno de los atributos utilizados.
- Método *private void actionPerformed (java.awt.event.ActionEvent evt)*, el cual define las acciones que se realizarán cada vez que se pulse algunos de los botones de la ventana



- **DecompresorZip**

Clase que se invoca cuando queremos abrir un curso desde fichero. Con ella se consigue que el archivo .zip que se va a abrir se descomprima para obtener los archivos que posteriormente se utilizarán.

- **DefinirPlantilla**

Esta clase define la ventana que le aparecerá al usuario cuando define una plantilla para el nuevo tipo de tarea creada.



*Figura 125 Ventana para definir la plantilla de los nuevos tipos de tarea*

Tiene la misma estructura que la clase PantallaInicial. Dicha estructura se explica a continuación.

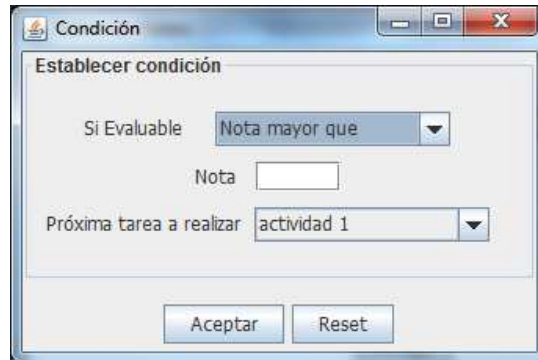
- Definición de los atributos que se utilizarán
- Constructor que hace visible la ventana e invoca al método initComponents()
- Método private void initComponents() que inicializa todos y cada uno de los atributos utilizados.
- Método private void actionPerformed (java.awt.event.ActionEvent evt), el cual define las acciones que se realizarán cada vez que se pulse algunos de los botones de la ventana.

- **Duracion**

La clase Duración define las comprobaciones necesarias para comprobar que las fechas introducidas en la herramienta son correctas. Se comprueba si las fechas introducidas son correctas y si las fechas de inicio son anteriores a las de fin, etc.

- **EditarCondicionEditor**

Esta clase define la ventana que le aparecerá al usuario cuando se edita una condición previamente añadida en el editor gráfico.



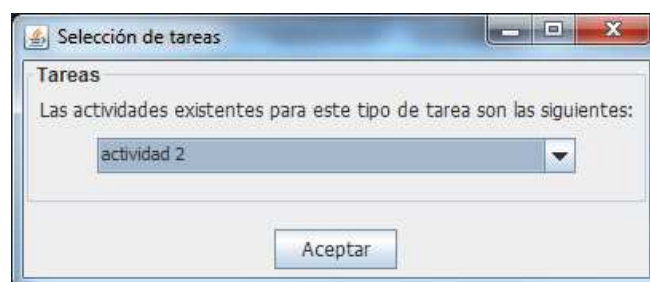
*Figura 126 Ventana para establecer una condición en la secuenciación de una unidad didáctica*

Tiene la misma estructura que la clase PantallaInicial. Dicha estructura se explica a continuación.

- Definición de los atributos que se utilizarán
- Constructor que hace visible la ventana e invoca al método initComponents()
- Método private void initComponents() que inicializa todos y cada uno de los atributos utilizados.
- Método private void actionPerformed (java.awt.event.ActionEvent evt), el cual define las acciones que se realizarán cada vez que se pulse algunos de los botones de la ventana.

- **EditarEditor**

Esta clase define la ventana que le aparecerá al usuario cuando se edita una caja de tarea en el editor gráfico, la cual permitirá asignar el nombre concreto de la tarea a la caja.



*Figura 127 Ventana para editar una tarea en el editor gráfico*

Su estructura es la siguiente:

- Definición de los atributos que se utilizarán
- Constructor que hace visible la ventana e invoca al método initComponents()
- Método `private void initComponents()` que inicializa todos y cada uno de los atributos utilizados.
- Método `private void actionPerformed (java.awt.event.ActionEvent evt)`, el cual define las acciones que se realizarán cada vez que se pulse algunos de los botones de la ventana.

- **EditarUnionEditor**

Esta clase define la ventana que le aparecerá al usuario cuando se edita una caja de unión previamente añadida en el editor gráfico.



*Figura 128 Ventana para editar una caja de tipo Unión en el editor gráfico*

La estructura de esta clase es la siguiente:

- Definición de los atributos que se utilizarán
- Constructor que hace visible la ventana e invoca al método initComponents()
- Método `private void initComponents()` que inicializa todos y cada uno de los atributos utilizados.
- Método `private void actionPerformed (java.awt.event.ActionEvent evt)`, el cual define las acciones que se realizarán cada vez que se pulse algunos de los botones de la ventana.

- **EditorGrafico**

Esta clase define la ventana que implementa el editor gráfico así como toda su funcionalidad (movilidad de cajas, unión de las cajas mediante flechas, etc.)

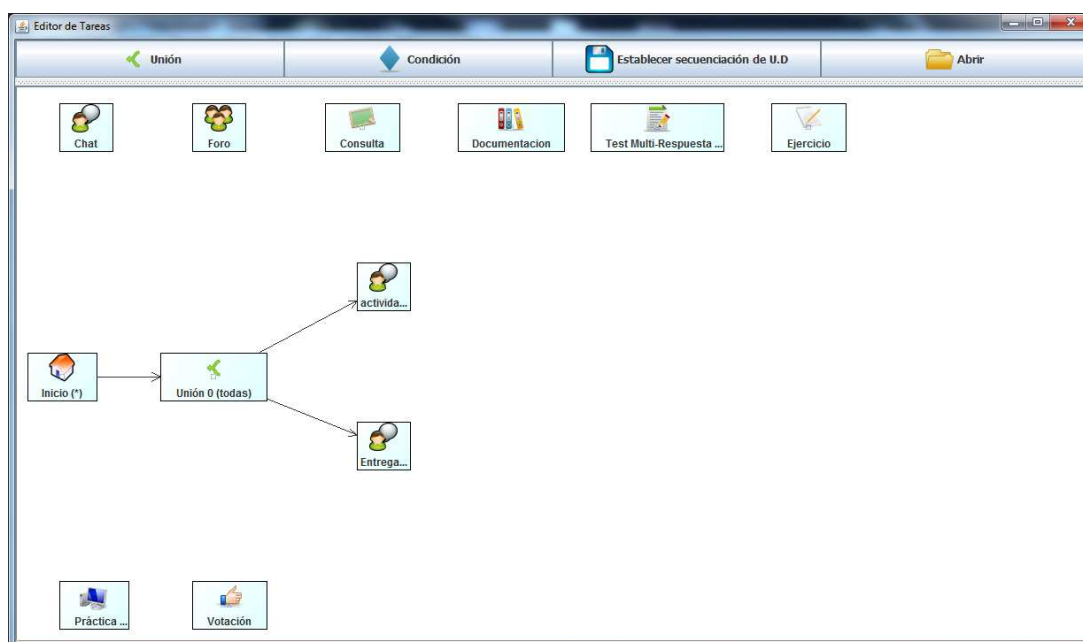


Figura 129 Editor gráfico

Esta clase consta de:

- Constructores: Métodos que se invocan para crear la ventana, así como cargar las tareas definidas para cada unidad didáctica. Dependiendo de si el constructor se invoca por primera vez, o se intenta abrir un gráfico desde fichero, se invocará a uno u otro.
- Método **public void** saveFile(), el cual permite guardar en un fichero XML la información extraída de la secuenciación llevada a cabo en el editor gráfico.
- Métodos **public void** openFile(String ruta) y **public void** openFile(), los cuales permiten cargar el gráfico, o bien desde un archivo cuyo path se pasa como parámetro o bien mediante un diálogo que permite al usuario cargar el gráfico deseado.
- Métodos **public void** insertCellInicio(Point2D point, String nombre), **public void** insertCellUnion(Point2D point, String nombre), **public void** insertCellCondicion(Point2D point, String nombre), **public void** insertCell(Point2D point, String nombre) los cuales permiten insertar celdas de inicio, unión, condición o tarea de cualquier tipo respectivamente al editor gráfico.
- Método **public void** recorrerProfundidadIterativo(Vector cajas\_rec, CajasEditor nodoInicio), el cual permite recorrer el grafo en profundidad de forma iterativa. De esta manera se van añadiendo todas las cajas y flechas para reconstruir el grafo guardado.
- Método **public Map** createCellAttributes(Point2D point, String nombre), el cual define los atributos (color, imagen etc.) que tendrá cada una de las cajas del editor gráfico.
- Método **public void** connect(Port source, Port target, Point2D firstPort, Point2D secondPort). Es uno de los métodos más importantes de la clase. Este método permite conectar o no, las distintas cajas del editor, realizando las comprobaciones necesarias para ello.

- Método `public Map createEdgeAttributes()`. Este método define las características de las flechas de unión.

Además esta clase consta de varias subclases:

- `MyEdgeHandle`: Define los métodos necesarios para el manejo de edges. Permite añadir/eliminar puntos hacia o desde un borde.
- `MyModel`: Define un modelo personalizado que no permite filtrar el tipo de conexión que se permite en el gráfico.
- `MyMarqueeHandler`: Define un `MarqueeHandler` personalizado que conecta vértices, menús emergente `PopupMenu` y eventos de ratón.
- `ManejoBotones`: Define los botones que aparecen en el editor gráfico, tanto su apariencia como su funcionalidad.

## • **EntradaSalida**

Esta clase proporciona toda la funcionalidad necesaria para guardar y leer la información introducida en la herramienta en un fichero de texto.

Para ello, posee varios métodos para escribir en fichero:

- `public void escribirInfoGeneral(String titulo, String descripcion, String audiencia, String prerequisites)`
- `public void escribirDuracion(String nh, String di, String mi, String ai, String df, String mf, String af, String nc)`
- `public void escribirObjetivos(Vector filas_obj)`
- `public void escribirUnidadesDidacticas(Vector unidades)`
- `public void escribirTiposTareas(Vector tipos_tareas)`
- `public void escribirTareas(Vector ser_tarea, Vector v_soporte, Vector ser_tipo)`
- `public void escribirPorSerDeUnTipo(String id, Vector ser_tipo)`
- `public void escribirPorSerEvaluable(TareaGenerica tg)`
- `public void escribirRolesExistentes(Vector roles)`
- `public void escribirTareasSoporte(Vector tareas_soporte)`
- `public void escribirCondiciones(Vector condiciones)`

Tal y como sus nombres indican, escriben en el fichero la información que se indica.

Por otra parte la clase posee el método:

```
public void leerDeFichero(PantallaInicial p, String fichero)
```

Este método es llamado cuando queremos cargar un curso previamente creado en la herramienta. Este método es el encargado de la lectura del fichero, así como de la llamada a los métodos necesarios para la introducción de información en el mismo.

- **FlechasEditor**

La clase FlechasEditor define los atributos que posee una flecha:

- *String origen*: Especifica el origen de la flecha.
- *String destino*: Especifica el destino de la flecha.

Esta clase cuenta con un constructor y los métodos *get* y *set* correspondientes.

- **GuardarInformacion**

Esta clase es la encargada de crear varios documentos XML. Los documentos XML creados a partir de esta clase son los siguientes.

- Un documento compatible IMS-LD.
- Un documento que contiene información general del curso
- Un documento que contiene la información sobre las unidades didácticas.
- Un documento que contiene los tipos de tareas y sus correspondientes campos, existentes para el curso.
- Un documento que contiene la información de las tareas (información por pertenecer a un tipo de tarea, si es evaluable, si es entregable, etc.)

Toda la información contenida en este documento se obtiene de las pantallas de la aplicación, y se almacena en los ficheros XML utilizando la librería JDOM.

En la parte superior de la clase se definen todos los elementos necesarios para la creación de los ficheros XML, el constructor que crea la estructura del fichero XML genérico, y luego todos los métodos encargados de toda la información necesaria.

- **InfTareaPorSerTarea**

Esta clase define la ventana que aparece automáticamente cuando se añade una tarea en la pestaña Tareas, con el fin de permitir al usuario introducir información adicional sobre la tarea en cuestión. La clase implementa la primera pantalla (por ser tarea), y la segunda pantalla (por ser de un tipo) y todas las pantallas posteriores que se necesiten dependiendo de si la tarea es evaluable, entregable, o ambas (pantallas por ser evaluable y por ser entregable)

Figura 130 Ventana para añadir información a la tarea

La estructura de la clase es la siguiente:

- Definición de los atributos que se utilizarán
- Constructor que hace visible la ventana e invoca al método initComponents()
- Método `private void initComponents()` que inicializa todos y cada uno de los atributos utilizados.
- Métodos `private void actionPerformedX (java.awt.event.ActionEvent evt)`, donde X es un número que pertenece a cada una de las ventanas. Estos métodos definen las acciones que se realizarán cada vez que se pulse algunos de los botones de cada una de las ventanas.

## • Mensaje

Esta clase define la ventana que le aparecerá para mostrarle al usuario un mensaje de error o advertencia.

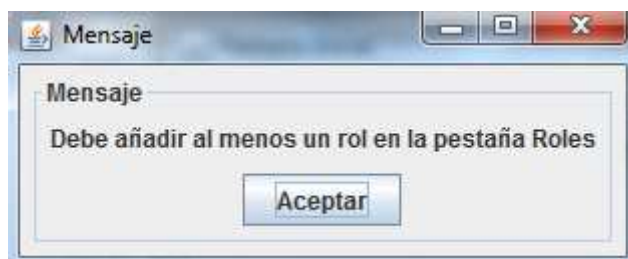


Figura 131 Mensaje informativo o de error

Su estructura es la siguiente:

- Definición de los atributos que se utilizarán
- Constructor que hace visible la ventana e invoca al método initComponents()
- Método private void initComponents() que inicializa todos y cada uno de los atributos utilizados.
- Método private void actionPerformed (java.awt.event.ActionEvent evt), el cual define las acciones que se realizarán cada vez que se pulse algunos de los botones de la ventana.

- **MensajeErrores**

Esta clase define la ventana que se mostrará al usuario cuando este pulse el botón “Aceptar” de la pestaña UD-Tareas, siempre y cuando se haya cometido algún error al rellenar la información del curso. El mensaje que se mostrará será un listado de todos los errores cometidos tal y como se puede observar en la Figura 132.



*Figura 132 Mensaje con todos los errores producidos en la creación del curso*

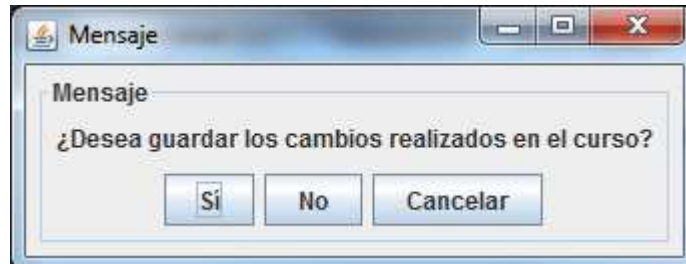
Su estructura es la siguiente:

- Definición de los atributos que se utilizarán
- Constructor que hace visible la ventana e invoca al método initComponents()
- Método private void initComponents() que inicializa todos y cada uno de los atributos utilizados.
- Método private void actionPerformed (java.awt.event.ActionEvent evt), el cual define las acciones que se realizarán cada vez que se pulse algunos de los botones de la ventana.



- **MensajeSalir**

Esta clase implementa la ventana que aparecerá cuando el usuario pulse o bien Archivo/Salir desde el menú o bien el botón de aspa de la parte superior derecha de la aplicación. Esta ventana sirve como comprobación de que el usuario realmente desea salir, y también da la oportunidad de guardar la información introducida antes de salir de la herramienta si no se había hecho antes.



*Figura 133 Mensaje para salir de la aplicación*

Su estructura es la siguiente:

- Definición de los atributos que se utilizarán
- Constructor que hace visible la ventana e invoca al método initComponents()
- Método private void initComponents() que inicializa todos y cada uno de los atributos utilizados.
- Método private void actionPerformed (java.awt.event.ActionEvent evt), el cual define las acciones que se realizarán cada vez que se pulse alguno de los botones de la ventana.

- **ModeloListaEditor**
- **ModeloListaEvaluable**
- **ModeloListaRoles**
- **ModeloListaSoporte**
- **ModeloListaTipoTarea**
- **ModeloListaUnidades**

Todas estas clases, tal y como su propio nombre indican, definen los modelos de las listas que se utilizan en la herramienta: listas del editor, de tareas evaluables, de roles, de tareas soporte, tipos de tareas y de unidades didácticas.

Todas ellas disponen de un constructor para poder crear una instancia de ellas, y de métodos get que permitan recuperar los parámetros y métodos que permiten añadir objetos a las listas.

- **ModeloTablaObjetivos**
- **ModeloTablaPlantilla**
- **ModeloTablaRoles**
- **ModeloTablaTareas**

- **ModeloTablaTipoTareas**
- **ModeloTablaUnidades**

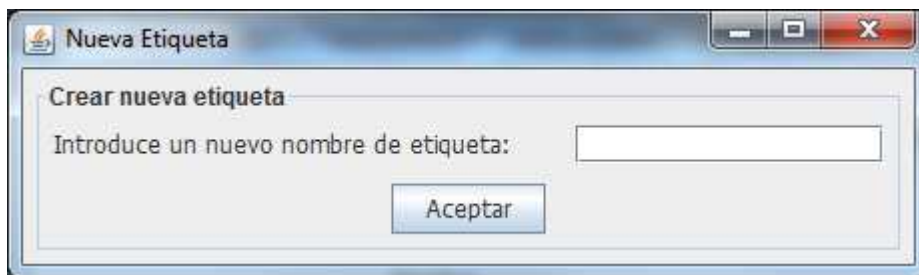
Estas clases definen los modelos de las tablas existentes en la herramienta: objetivos, plantilla, roles, tareas, tipos de tareas y unidades didácticas.

La estructura de todas ellas es la siguiente:

- Un constructor que hace una llamada al método `cargarDatos()`
- Un método `cargarDatos()` que añade las columnas que posee cada una de las tablas.
- Métodos `get` y `set` que permiten recuperar y establecer datos como el nombre de las columnas o el número de columnas o filas.
- Un método `public void aniadir(...)` donde los puntos suspensivos representan los parámetros que recibe el método. Este método sirve para añadir una nueva fila a la tabla.
- Un método `public void aniadirLeer(...)` donde los puntos suspensivos representan los parámetros que recibe el método. Este método permite añadir las columnas a la tabla cuando las estamos cargando desde fichero. Por lo tanto es llamado cuando se está leyendo el documento de texto del que se extrae la información guardada en una ejecución anterior.
- Un método `public void eliminar(int fila)` que permite eliminar de la tabla la fila que se le pasa como parámetro.

- **NuevaEtiquetaRol**

Esta clase implementa la ventana que aparecerá cuando el usuario quiera crear una nueva etiqueta para un rol seleccionando la opción 'Crear nueva etiqueta' en el `jComboBox` de la pestaña Roles.



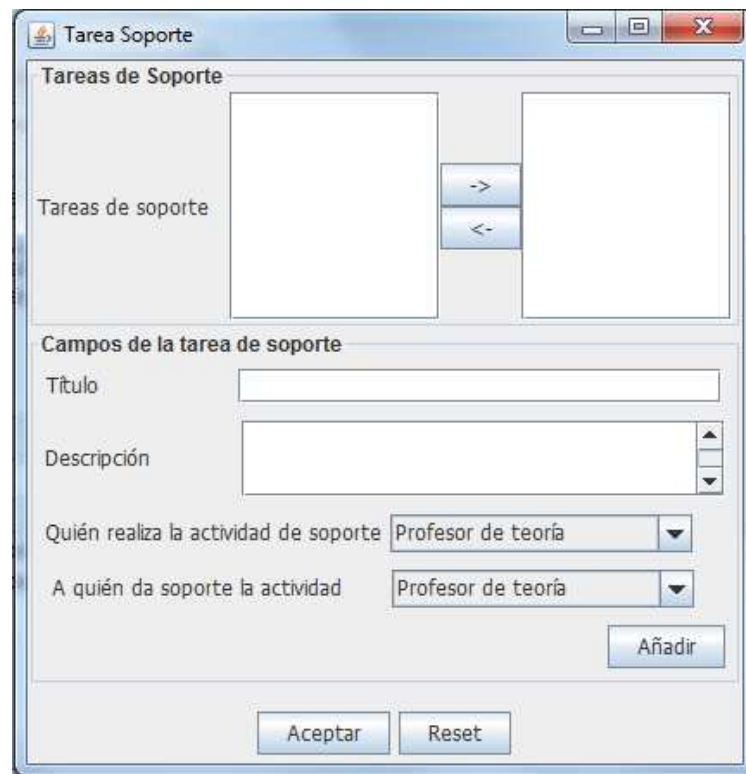
*Figura 134 Ventana para crear una nueva etiqueta de rol*

Su estructura es la siguiente:

- Definición de los atributos que se utilizarán
- Constructor que hace visible la ventana e invoca al método  `initComponents()`
- Método `private void initComponents()` que inicializa todos y cada uno de los atributos utilizados.
- Método `private void actionPerformed (java.awt.event.ActionEvent evt)`, el cual define las acciones que se realizarán cada vez que se pulse alguno de los botones de la ventana.

- **NuevaTareaSoporte**

Esta clase implementa la ventana que aparecerá cuando el usuario cree una actividad de soporte desde la ventana de información adicional de una tarea.



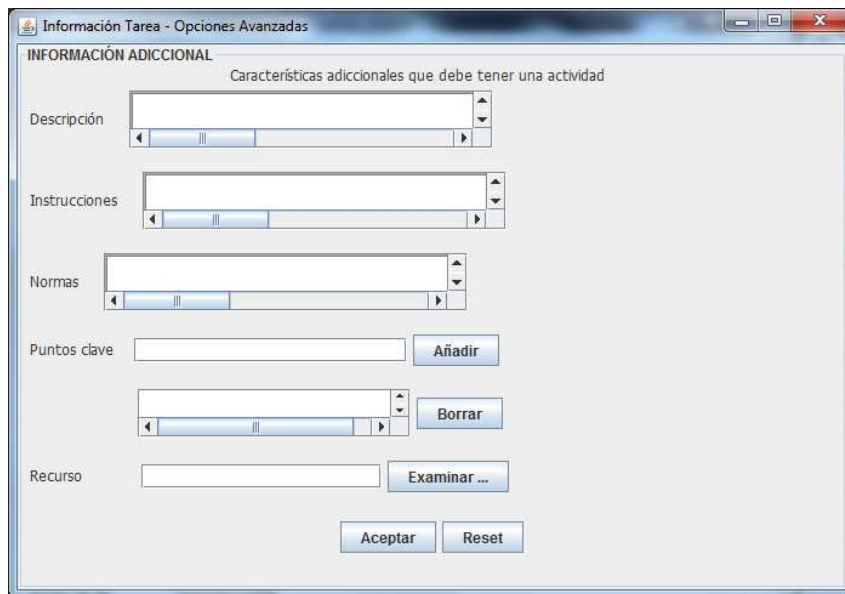
*Figura 135 Crear nueva tarea soporte.*

Su estructura es la siguiente:

- Definición de los atributos que se utilizarán
- Constructor que hace visible la ventana e invoca al método initComponents()
- Método `private void initComponents()` que inicializa todos y cada uno de los atributos utilizados.
- Método `private void actionPerformed (java.awt.event.ActionEvent evt)`, el cual define las acciones que se realizarán cada vez que se pulse alguno de los botones de la ventana.

- **OpcionesAvanzadas1**

Esta clase implementa la ventana que aparecerá cuando el usuario quiere introducir información adicional de una tarea por ser de un tipo. La ventana aparecerá cuando se pulsa el botón *Opciones Avanzada*.



*Figura 136 Introducción de opciones avanzadas*

Su estructura es la siguiente:

- Definición de los atributos que se utilizarán
- Constructor que hace visible la ventana e invoca al método  `initComponents()`
- Método `private void initComponents()` que inicializa todos y cada uno de los atributos utilizados.
- Método `private void actionPerformed (java.awt.event.ActionEvent evt)`, el cual define las acciones que se realizarán cada vez que se pulse alguno de los botones de la ventana.
- Otros métodos adicionales que permiten cargar un campo u otro, dependiendo del tipo de tarea que sea.

#### • PilaGrafo

Esta clase es una clase auxiliar que representa una pila que se utiliza para almacenar las cajas visitadas cuando se recorre el grafo para reconstruirlo desde fichero.

La clase consta de tres métodos que se explican a continuación.

- `public void addNodo(CajasEditor nodo)`. Este método añade un nodo a la pila cuando éste es visitado.
- `public CajasEditor getNodo()`. Este método nos devuelve la caja visitada.
- `public void mostrarContenido()`. Este método muestra el contenido de la caja visitada.

#### • PorSerDeUnTipo

Esta clase define los atributos que posee una tarea por ser de un tipo. Dichos atributos son los siguientes.

- *String id*: Identificador de la tarea
- *String tipo*: Tipo al que pertenece la tarea.
- *Vector campos*: Vector que contiene los campos a rellenar para esa tarea en concreto.

Esta clase cuenta con dos constructores y los métodos *get* y *set* correspondientes.

- **RenderTabla**
- **RenderTablaBorrar**

Estas clases se utilizan para personalizar la forma en que se presentan los datos de los botones Editar y Borrar de todas las tablas de la herramienta.

- **TareaEditor**

Esta clase define los atributos que poseerá una tarea del editor gráfico, de forma que se permita unir distintos atributos en un mismo elemento. Los atributos definidos en esta clase son:

- *String identificador*: Identificador de la tarea
- *String tipo*: Tipo de tarea
- *String titulo*: Nombre de la tarea
- *Vector roles*: Vector que contendrá los roles que serán los encargados de llevar a cabo la tarea
- *Vector tareas\_s*: Vector que contendrá las tareas soporte definidas para la tarea
- *boolean borrado*: Boolean que indica si la tarea ha sido o no borrada.

- **TareaGenerica**

Esta clase define los atributos que poseerá una tarea del editor gráfico, de forma que nos permita unir distintos atributos en un mismo elemento. Los atributos definidos en esta clase son:

- *String identificador*: Identificador de la tarea
- *String titulo\_tarea*: Nombre de la tarea
- *String ud\_tarea*: Unidad didáctica a la que pertenece la tarea
- *String tipo\_tarea*: Tipo de tarea
- *String duración\_tarea*: La duración que va a tener la tarea
- *String descripción\_tarea*: Breve explicación de en qué consiste la tarea
- *String descripción\_exam\_tarea*: Breve explicación de en qué consiste una tarea evaluable.
- *boolean evaluable*: Indica si la tarea es o no evaluable
- *boolean entregable\_tarea*: Indica si la tarea es o no entregable.
- *boolean completa*: Indica si la tarea ha sido o no completada.

Esta clase cuenta con un constructor y los métodos *get* y *set* correspondientes.

- **TareaSoporte**

Esta clase define los atributos que poseerá una tarea de soporte, de forma que se permita unir distintos atributos en un mismo elemento. Los atributos definidos en esta clase son:

- *String identificador*: Identificador de la tarea soporte.
- *String nombre*: Nombre de la tarea soporte.
- *String descripción*: Breve explicación de en qué consiste la tarea soporte.
- *String quien\_realiza*: Roles que realizan la actividad de soporte.
- *boolean a\_quien\_soporta*: Roles sobre los que recae la actividad de soporte.

Esta clase cuenta con un constructor y los métodos *get* y *set* correspondientes.

- **TipoNuevo**

Esta clase, que extiende de la clase *TareaGenerica*, define los atributos que poseerá un tipo nuevo de tarea creado por el usuario, de forma que se permita unir distintos atributos en un mismo elemento. Los atributos definidos en esta clase son:

- *String nombre\_ttarea*: Identificador del nuevo tipo de tarea.
- *String descripción\_ttarea*: Breve explicación de en qué consiste el nuevo tipo de tarea.
- *String iniciales\_ttarea*: Iniciales asignadas al nuevo tipo de tarea.
- *String servicio\_ttarea*: Servicio que presta el nuevo tipo de tarea.
- *Vector campos\_ttarea*: Este vector contendrá los campos que se definirán para el nuevo tipo de tarea.

Esta clase cuenta con un constructor y los métodos *get* y *set* correspondientes.

- **UnidadDidactica**

Esta clase define los atributos que poseerá una unidad didáctica, de forma que se permita unir distintos atributos en un mismo elemento. Los atributos definidos en esta clase son:

- *String id*: Identificador de la unidad didáctica.
- *String nombre*: Nombre asignado a la unidad didáctica.
- *String descripcion*: Breve explicación de en qué consiste la unidad didáctica.
- *String duracion*: Duración de la unidad didáctica.
- *Vector objetivos*: Este vector contendrá los objetivos que se pretenden alcanzar con la unidad didáctica definida.

Esta clase cuenta con un constructor y los métodos *get* y *set* correspondientes.

- **VectorErrores**

Esta clase define un vector que contendrá todos los errores de introducción de información realizada por el usuario. Su contenido será el que se mostrará cuando el usuario pulse el botón Aceptar de la última pestaña: Ud-Tareas.

La clase contiene los métodos get y set para obtener e introducir un error cometido en el Vector, así como un método (`public static String muestraErrores()`) que mostrará los errores de forma que haya un salto de línea al final de cada error.

CAPÍTULO: Tutorial para la realización de modificaciones en la herramienta.



## Anexo 2.

### Enunciado del curso de prueba creado para la evaluación de la herramienta.

El objetivo de este enunciado es la creación de un curso que sigue la especificación IMS Learning Design a través de la herramienta CourseEditor.

Para ello, nos basaremos en la programación de la asignatura cuatrimestral 'Programación de Sistemas' que sigue el plan Bolonia, consistente en 14 sesiones de teoría y 14 sesiones de prácticas.

En la primera pestaña de la herramienta, Info. General, se debe rellenar la siguiente información:

- **Título:** Programación de Sistemas
- **Descripción:** Asignatura correspondiente a la rama de Ingeniería y arquitectura impartida por el Departamento de Ingeniería Telemática.
- **Audiencia:** Alumnos de primer curso de Grado en Ingeniería Telemática, Ingeniería de Sistemas Audiovisuales e Ingeniería de Sistemas de Comunicaciones.
- **Prerrequisitos:** Haber cursado la asignatura 'Programación'
-

En la segunda pestaña, Duración, se debe rellenar la siguiente información:

- **Número de horas:** 42
- **Fecha Inicio:** 28/01/2013
- **Fecha de fin:** 10/05/2013
- **Número de créditos:** 6

En la tercera pestaña, Objetivos, se añaden los objetivos de la asignatura, de forma que la suma de todos sus pesos sume 1. Los objetivos para esta asignatura serán:

- Objetivo 1: Interpretar, explicar y ejemplificar los mecanismos de la POO. Programar aplicaciones orientadas a objetos
  - o Peso: 0.1
- Objetivo 2: Interpretar, explicar y ejemplificar los conceptos de recursión y complejidad. Representar, diseñar e implementar algoritmos recursivos.
  - o Peso: 0.1
- Objetivo 3: Interpretar, explicar y ejemplificar los fundamentos de las estructuras de datos más habituales tales como listas y árboles
  - o Peso: 0.2
- Objetivo 4: Interpretar, explicar y ejemplificar los distintos algoritmos de trabajo sobre estructuras de datos. Implementar algoritmos y usar algunas estructuras de datos importantes (listas enlazadas, pilas, colas, colas dobles, árboles)
  - o Peso: 0.2
- Objetivo 5: Interpretar, explicar y ejemplificar los fundamentos de programación de interfaces gráficas. Implementar GUIs sencillas.
  - o Peso: 0.1
- Objetivo 6: Ser capaz de crear, compilar y ejecutar un programa en java utilizando un entorno de desarrollo integrado
  - o Peso: 0.1
- Objetivo 7: Explicar y presentar de forma concisa el trabajo realizado (técnicas, estructuras de datos, problemas encontrados, soluciones aportadas) que haya sido previamente consensuado en el grupo
  - o Peso: 0.1
- Objetivo 8: Ejemplificar y aplicar estrategias para resolver los típicos problemas de trabajo en grupo
  - o Peso: 0.1

En la cuarta pestaña, U.D, se definen las siguientes unidades didácticas.

- **Título:** Introducción y repaso
- **Duración:** 90  
**Descripción:** Presentación del curso (temario, metodología, evaluación, etc.); Instalación y configuración del entorno de trabajo; Repaso de conceptos básicos de programación

- **Título:** Ejercicios de repaso
- **Duración:** 90  
**Descripción:** Uso del entorno de desarrollo; Uso del depurador; Ejercicios de repaso
- **Título:** Orientación a Objetos y Herencia
- **Duración:** 90  
**Descripción:** Concepto de clase y objeto; Herencia; Sobreescritura y sobrecarga; Ocultación de la información; Modificadores de acceso; Constructores
- **Título:** Ejercicios de Orientación a Objetos y Herencia. Presentación enunciado Proyecto M1
- **Duración:** 90  
**Descripción:** Ejercicios
- **Título:** Orientación a Objetos e Interfaces
- **Duración:** 90  
**Descripción:** Clases abstractas; Interfaces; Herencia múltiple; Polimorfismo
- **Título:** Ejercicios de Orientación a Objetos e Interfaces
- **Duración:** 90  
**Descripción:** Ejercicios
- **Título:** Listas enlazadas y doblemente enlazadas
- **Duración:** 90  
**Descripción:** Definición; Tipos; Ejemplos de Implementación; Operaciones básicas (inserción, eliminación, búsqueda)
- **Título:** Ejercicios de Listas enlazadas simples y dobles
- **Duración:** 90  
**Descripción:** Ejercicios
- **Título:** Pilas y Colas
- **Duración:** 90  
**Descripción:** Definición; Tipos; Ejemplos de uso y de implementación; Operaciones básicas (inserción, eliminación)
- **Título:** Ejercicios de Pilas y Colas. Entrega del Proyecto M1. Presentación enunciado Proyecto M2-Pilas-colas
- **Duración:** 90  
**Descripción:** Ejercicios, presentación y entrega.
- **Título:** Examen parcial 1
- **Duración:** 90  
**Descripción:** Examen escrito (teoría + problemas)
- **Título:** Trabajo en el Proyecto

**Duración:** 90

**Descripción:** Resolución de dudas.

- **Título:** Recursión
- **Duración:** 90  
**Descripción:** Concepto; Tipos de recursión; Ejemplos de implementación y ejecución de métodos recursivos
- **Título:** Ejercicios de Recursión. Entrega del Proyecto M2  
Presentación enunciado Proyecto M3-Árboles  
**Duración:** 90  
**Descripción:** Ejercicios, presentación y entrega.
- **Título:** Árboles  
**Duración:** 90  
**Descripción:**  
Definición; Propiedades; Ejemplos de uso y de implementación; Recorridos (preorden, postorden, inorden); Árboles binarios
- **Título:** Ejercicios de Árboles I  
**Duración:** 90  
**Descripción:** Ejercicios
- **Título:** Árboles II  
**Duración:** 90  
**Descripción:** Árboles binarios de búsqueda; Ejemplos de uso y de implementación; Operaciones básicas (inserción, eliminación, búsqueda); Montículos; Ejemplos de uso y de implementación; Operaciones básicas (inserción, eliminación, búsqueda)
- **Título:** Ejercicios de Árboles II  
**Duración:** 90  
**Descripción:** Ejercicios
- **Título:** Interfaces Gráficas I  
**Duración:** 90  
**Descripción:** Definición; Conceptos básicos; Controles básicos; Diseño de ventanas. Ejemplos de implementación
- **Título:** Ejercicios de Interfaces Gráficas I  
Entrega del Proyecto M3  
**Duración:** 90  
**Descripción:** Ejercicios relacionados con interfaz para el proyecto
- **Título:** Interfaces Gráficas II  
**Duración:** 90  
**Descripción:** Modelo de eventos; Programación orientada a eventos; Ejemplos de implementación

- **Título:** Presentación del proyecto
- **Duración:** 90
- Descripción** Presentaciones
  
- **Título:** : Complejidad
- **Duración:** 90
- Descripción:** Definición; Ejemplos; Notación O; Complejidad vs Computabilidad
  
- **Título:** Ejercicios de Interfaces Gráficas II
- Indicaciones para el examen de laboratorio
- **Duración:** 90
- Descripción** Ejercicios relacionados con interfaz para el proyecto
  
- **Título:** Examen Parcial II
- **Duración:** 90
- Descripción** Examen escrito (teoría + problemas)
  
- **Título:** Resolución de examen de laboratorio
- **Duración:** 90
- Descripción** Resolución de problemas de los parciales realizados
  
- **Título :** Repaso
- **Duración:** 90
- Descripción:**  
Resolución de dudas; Resolución de problemas de examen de años anteriores ordinaria

En la quinta pestaña, T.Tareas, se definen los tipos de tareas que se utilizarán durante el curso. Para este curso utilizaremos: chat, foro, test-multirespuesta, ejercicio y práctica.

En la sexta pestaña, Roles, se definen los roles que formarán parte en el desarrollo del curso. Para ello, debemos rellenar la siguiente información para cada uno de los roles: nombre, descripción, pertenencia y etiqueta. Definiremos:

- **Nombre:** Profesor de teoría
- Descripción:** Profesor que se encargará de la docencia de los grupos magistrales
- Pertenencia:** Staff
- Etiquetar como:** Sin etiqueta
  
- **Nombre:** Profesor de prácticas
- Descripción:** Profesor que se encargará de la docencia de los grupos de prácticas
- Pertenencia:** Staff
- Etiquetar como:** Sin etiqueta
  
- **Nombre:** Alumnos
- Descripción:** Todos los alumnos matriculados en la asignatura

**Pertenencia:** Learner

**Etiquetar como:** Sin etiqueta

- **Nombre:** Alumnos de un grupo de practicas

**Descripción:** Alumnos que forman un grupo de prácticas

**Pertenencia:** Learner

**Etiquetar como:** Grupo de prácticas

En la séptima pestaña, Tareas, se definen las tareas de las que constará cada una de las unidades didácticas definidas anteriormente.

- **Introducción y repaso (Secuenciación 1)**
  - Chat: para alumnos y profesor de teoría
  - Consulta: para alumnos
  - Ejercicio: para alumnos.
- **Ejercicios de repaso (Secuenciación 2)**
  - Chat: para alumnos
  - Consulta: para alumnos y profesor de teoría
  - Documentación: para alumnos
  - Ejercicio: para alumnos. Evaluable y Entregable
- **Orientación a Objetos y Herencia (Secuenciación 3)**
  - Chat: para alumnos
  - Consulta: para alumnos y profesor de teoría
  - Documentación: para profesor de teoría
  - Ejercicio: para alumnos. Evaluable y Entregable
- **Ejercicios de Orientación a Objetos y Herencia (Secuenciación 4)**
  - Chat: para alumnos
  - Consulta: para alumnos y profesor de teoría
  - Documentación: para profesor de práctica
  - Ejercicio: para alumnos. Evaluable y Entregable
- **Orientación a Objetos e Interfaces (Secuenciación 5)**
  - Ejercicio: para alumnos.
  - Consulta: para alumnos y profesor de teoría
  - Documentación: para profesor de teoría
  - Chat: para alumnos
- **Ejercicios de Orientación a Objetos e Interfaces (Secuenciación 6)**
  - Ejercicio: para alumnos. Evaluable.
  - Chat: para alumnos y profesor de prácticas
  - Documentación: para profesor de prácticas
  - Consulta: para alumnos y profesor de prácticas
  - Votación: para alumnos.
- **Listas enlazadas y doblemente enlazadas (Secuenciación 7)**
  - Chat: para alumnos y profesor de prácticas

- Consulta: para alumnos y profesor de prácticas
- Documentación: para profesor de prácticas
- Ejercicio: para alumnos.
- Votación: para alumnos.
  
- **Pilas y Colas (Secuenciación 8)**
  - Chat: para alumnos y profesor de teoría
  - Consulta: para alumnos y profesor de teoría
  - Documentación: para profesor de teoría
  - Ejercicio: para alumnos.
  
- **Ejercicios de Pilas y Colas. Presentación enunciado Proyecto M2-Pilas-colas (Secuenciación 9)**
  - Chat: para alumnos y profesor de prácticas
  - Consulta: para alumnos y profesor de prácticas
  - Documentación: para profesor de prácticas
  - Ejercicio: para alumnos
  - Práctica: para alumnos
  
- **Recursión (Secuenciación 10)**
  - Ejercicio: para alumnos. Entregable y evaluable
  - Proyecto: para alumnos
  - Consulta: para alumnos y profesor de teoría
  - Chat: para alumnos y profesor de teoría
  - Documentación: para alumnos
  
- **Ejercicios de Recursión (Secuenciación 11)**
  - Chat: para alumnos y profesor de práctica.
  - Consulta: para alumnos y profesor de práctica
  - Documentación: para alumnos
  - Lectura: para alumnos
  
- **Árboles (Secuenciación 12)**
  - Chat: para alumnos y profesor de teoría
  - Ejercicio: para alumnos. Entregable y evaluable
  - Consulta: para alumnos y profesor de teoría
  - Lectura: para alumnos
  
- **Ejercicios de Árboles I (Secuenciación 13)**
  - Test-multirespuesta: para alumnos. Evaluable
  - Consulta: para alumnos y profesor de práctica
  - Compartir recurso: para alumnos
  - Chat: para alumnos y profesor de práctica
  - Lectura: para alumnos

- Ejercicio: para alumnos. Entregable
- **Árboles II (Secuenciación 14)**
  - Chat: para alumnos y profesor de teoría
  - Consulta: para alumnos y profesor de teoría
  - Ejercicio: para alumnos
  - Votación: para alumnos y profesor de teoría
  - Lectura: para alumnos
  - Caso de estudio: para alumnos y profesor de teoría
- **Ejercicios de Árboles II (Secuenciación 15)**
  - Chat: para alumnos y profesor de práctica
  - Ejercicio: para alumnos.
  - Lectura: para alumnos
  - Foro: para alumnos y profesor de práctica
  - Consulta: para alumnos y profesor de práctica
  - Práctica: para alumnos. Entregable
- **Interfaces Gráficas I (Secuenciación 16)**
  - Chat: para alumnos y profesor de teoría
  - Ejercicio: para alumnos
  - Votación: para alumnos
  - Caso de estudio: para alumnos
  - Consulta: para alumnos y profesor de teoría
  - Test-multirespuesta: para alumnos
  - Lectura: para alumnos
- **Ejercicios de Interfaces Gráficas I (Secuenciación 17)**
  - Ejercicio: para alumnos. Entregable
  - Test-multirespuesta: para alumnos. Evaluable
  - Práctica: para alumnos. Entregable
  - Chat: para alumnos y profesor de práctica
  - Consulta: para alumnos y profesor de práctica
  - Lectura: para alumnos.
  - Compartir recurso: para alumnos.
- **Interfaces Gráficas II (Secuenciación 18)**
  - Test-multirespuesta: para alumnos. Evaluable
  - Chat: para alumnos y profesor de teoría
  - Ejercicio: para alumnos. Entregable
  - Votación: para alumnos.
  - Consulta: para alumnos y profesor de teoría
  - Lectura: para alumnos.
  - Proyecto: para alumnos.



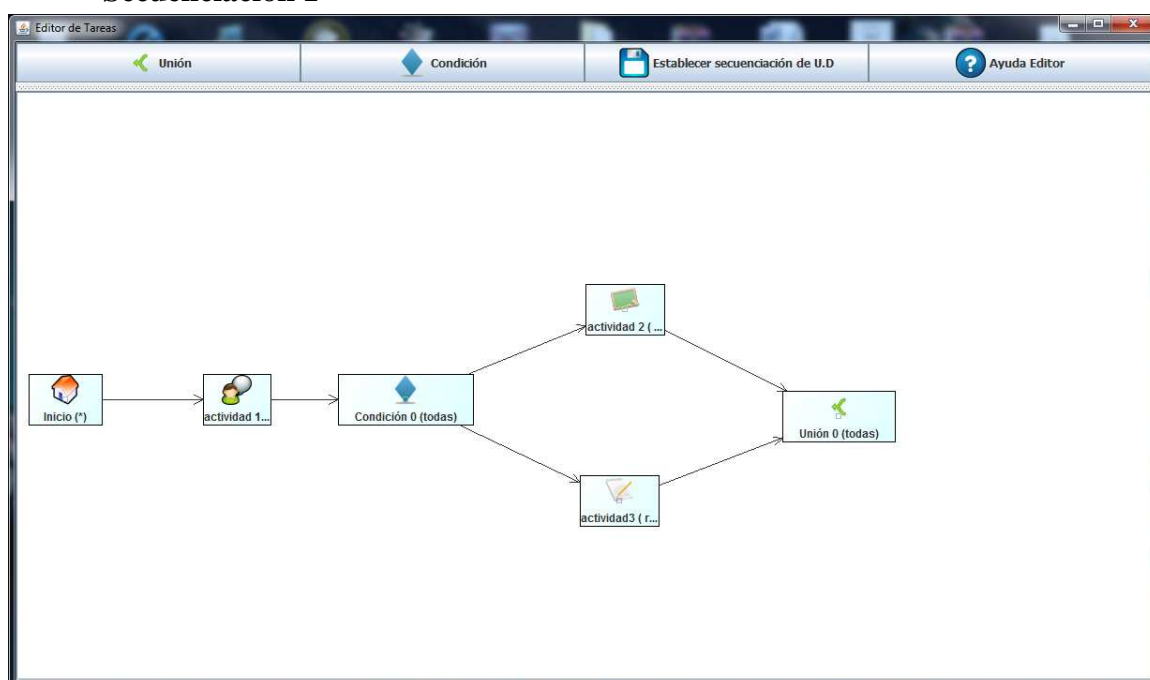
- **Complejidad (Secuenciación 19)**
  - Chat: para alumnos y profesor de teoría
  - Foro: para alumnos y profesor de teoría
  - Consulta: para alumnos y profesor de teoría
  - Lectura: para alumnos
  - Proyecto: para alumnos
  - Documentación: para alumnos y profesor de teoría
  - Ejercicio: para alumnos
  - Lectura: para alumnos
  
- **Ejercicios de Interfaces Gráficas II (Secuenciación 20)**
  - Chat: para alumnos y profesor de práctica
  - Foro: para alumnos y profesor de práctica
  - Documentación: para alumnos y profesor de práctica
  - Consulta: para alumnos y profesor de práctica
  - Lectura: para alumnos
  - Test-multirespuesta: para alumnos
  - Consulta: para alumnos y profesor de práctica
  - Ejercicio: para alumnos

CAPÍTULO: Enunciado del curso de prueba creado para la evaluación de la herramienta.

## Anexo 3.

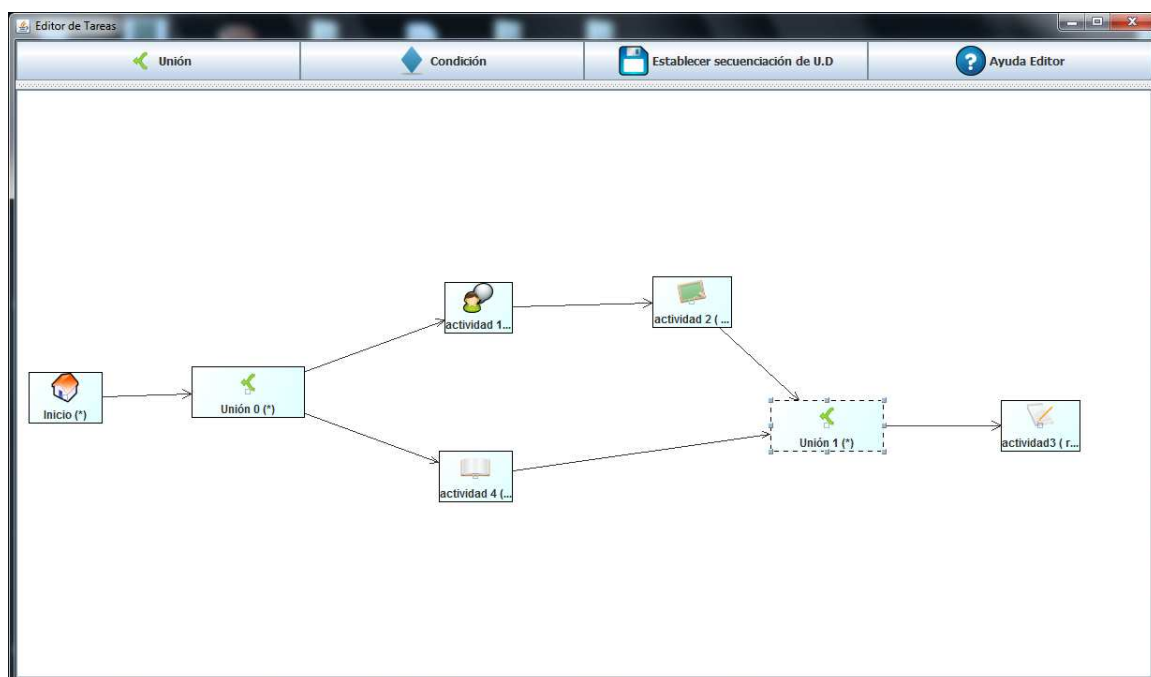
### Secuenciaciones del interfaz gráfico probadas.

- **Secuenciación 1**



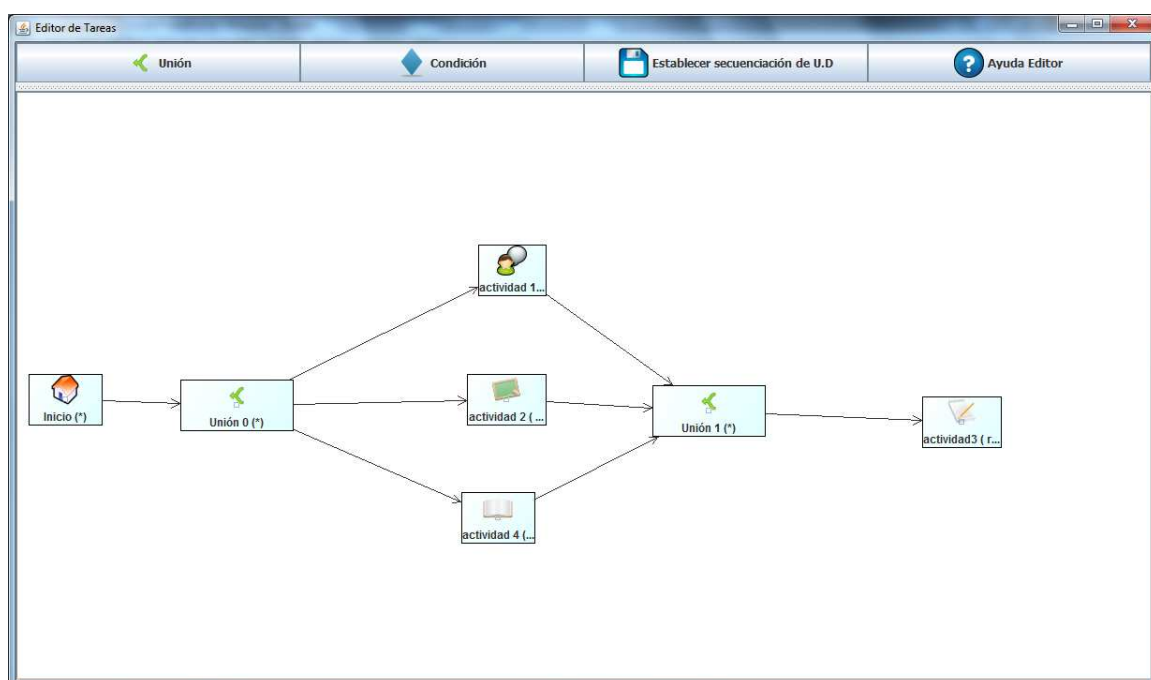
*Figura 137 Editor gráfico. Secuenciación 1*

- **Secuenciación 2**



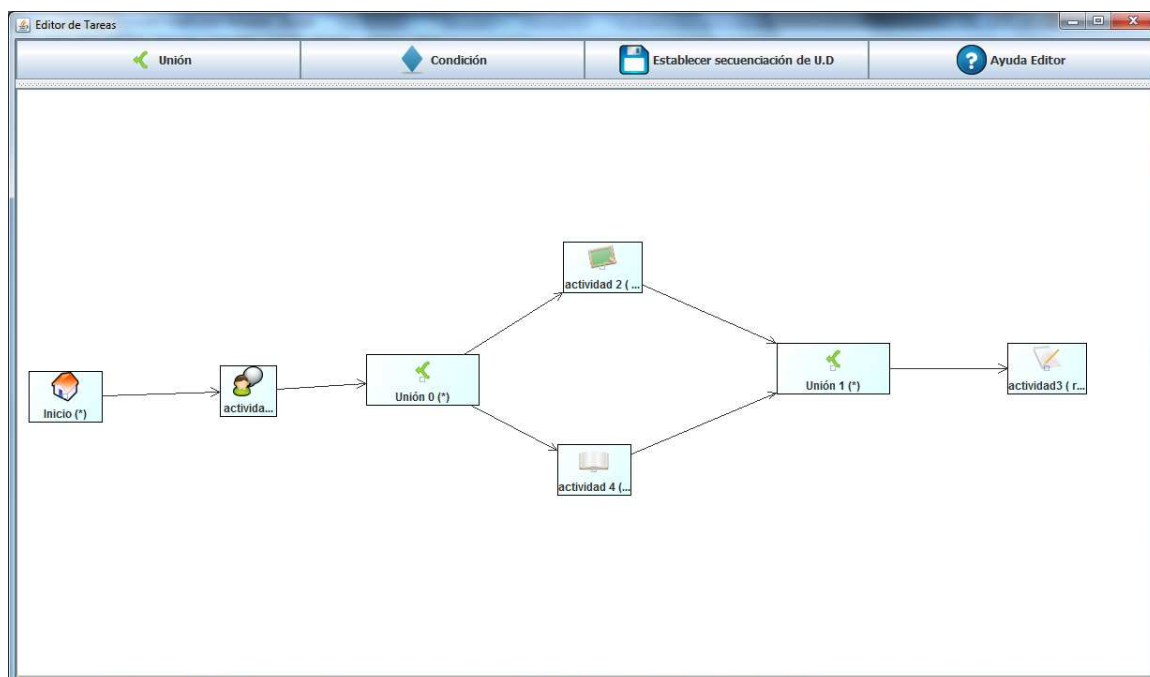
*Figura 138 Editor gráfico. Secuenciación 2*

- **Secuenciación 3**



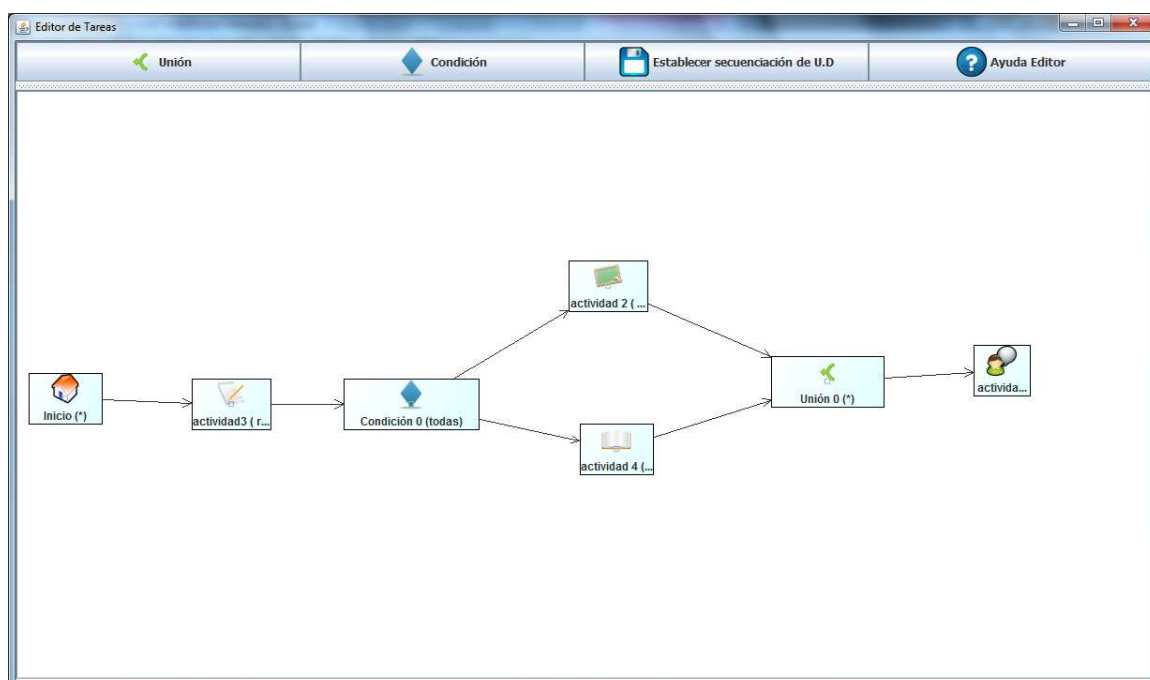
*Figura 139 Editor gráfico. Secuenciación 3*

- **Secuenciación 4**



*Figura 140 Editor gráfico. Secuenciación 4*

- **Secuenciación 5**



*Figura 141 Editor gráfico. Secuenciación 5*

- **Secuenciación 6**

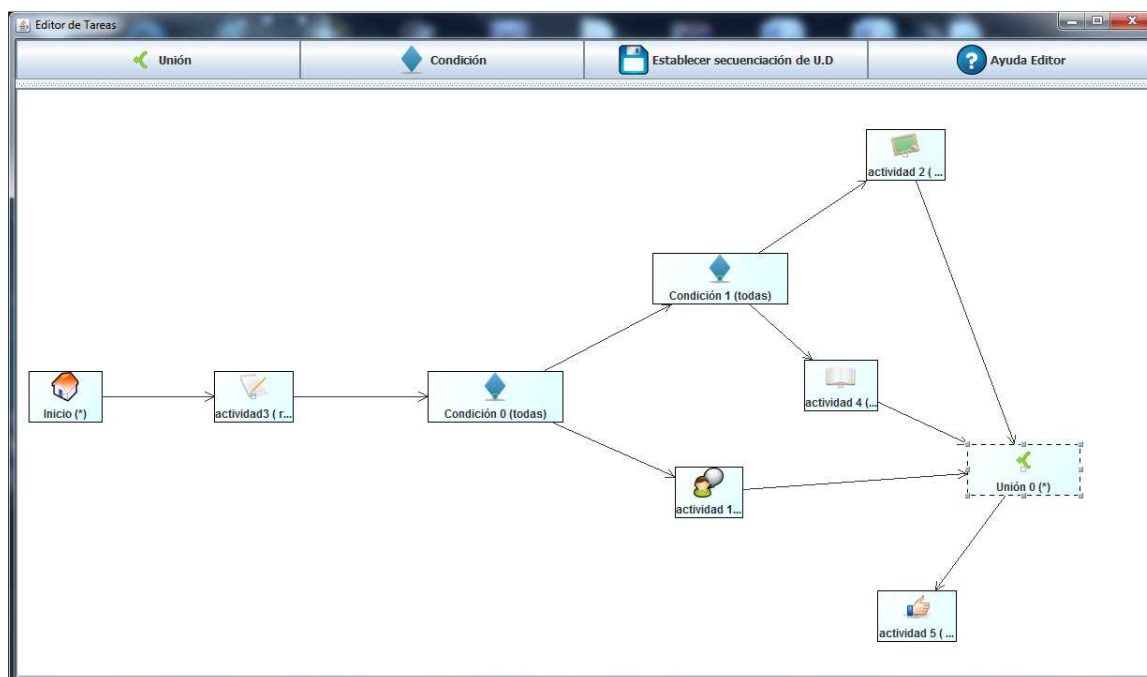


Figura 142 Editor gráfico. Secuenciación 6

- **Secuenciación 7**

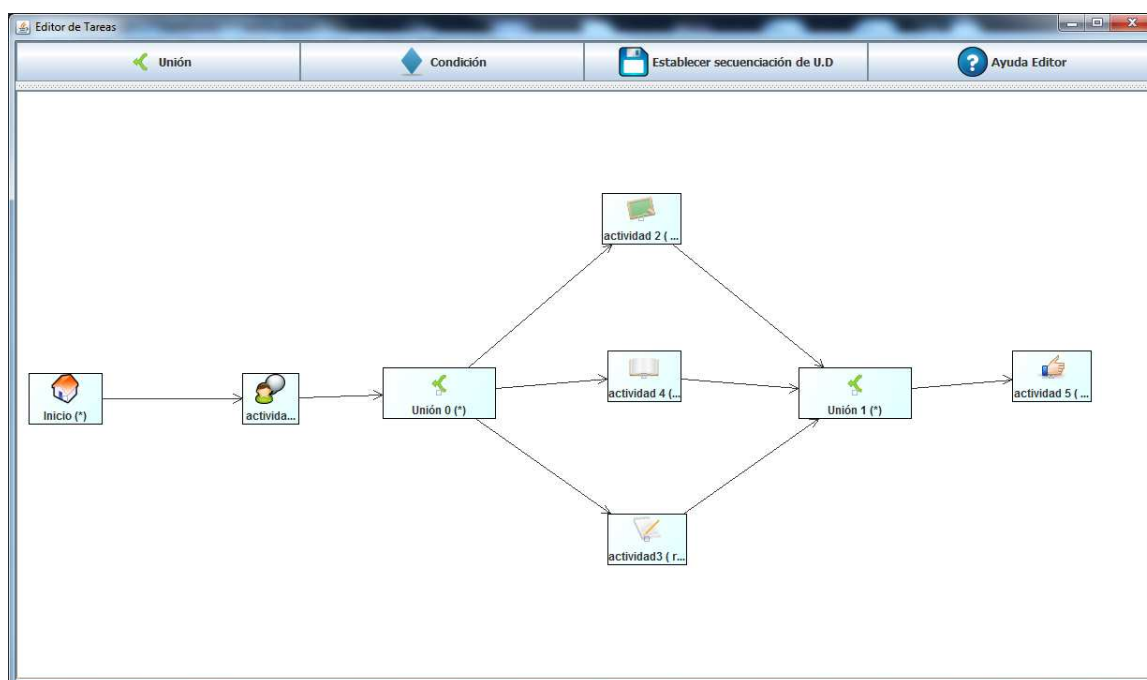


Figura 143 Editor gráfico. Secuenciación

- **Secuenciación 8**

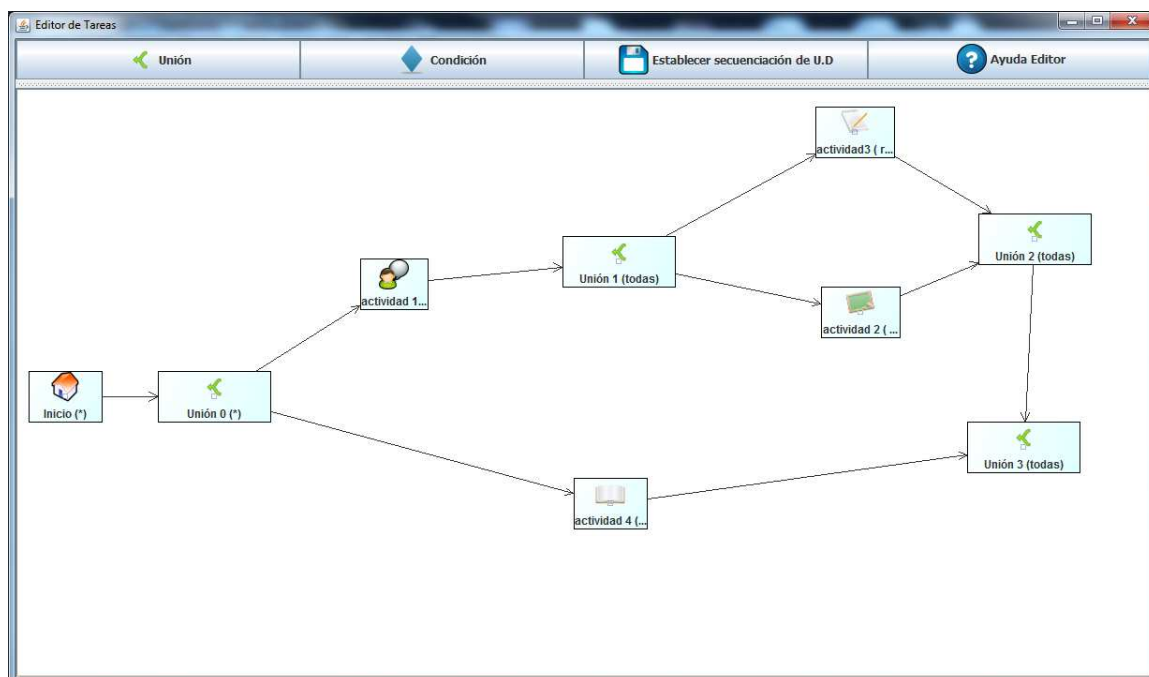


Figura 144 Editor gráfico. Secuenciación 8

- **Secuenciación 9**

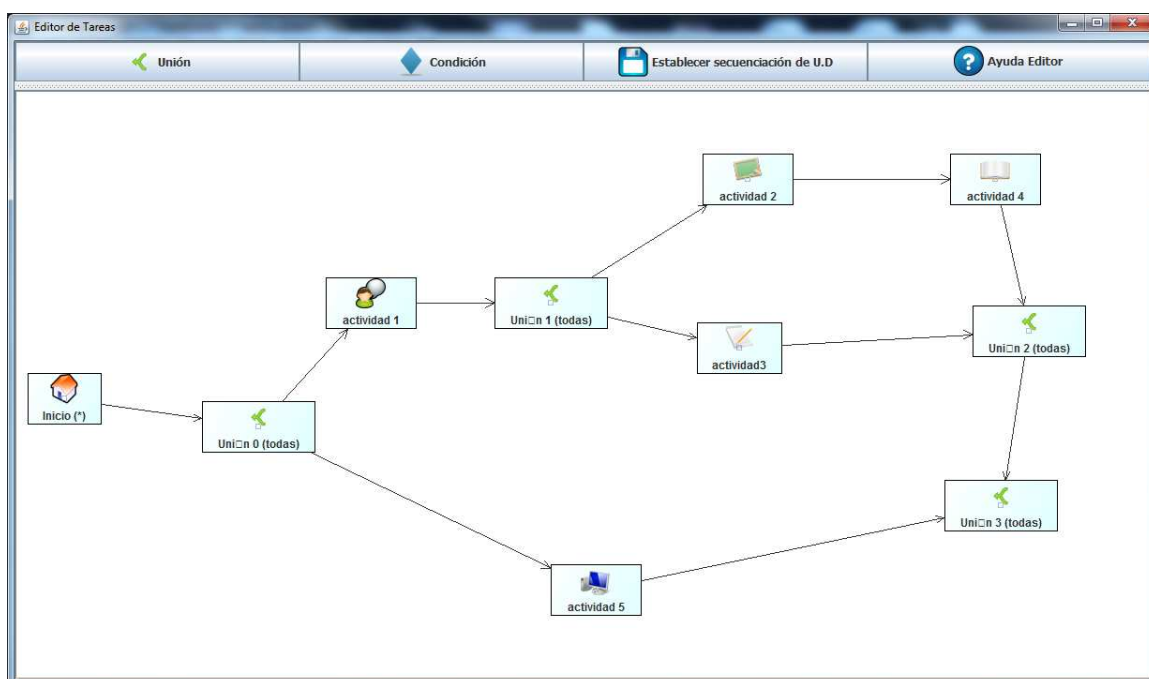
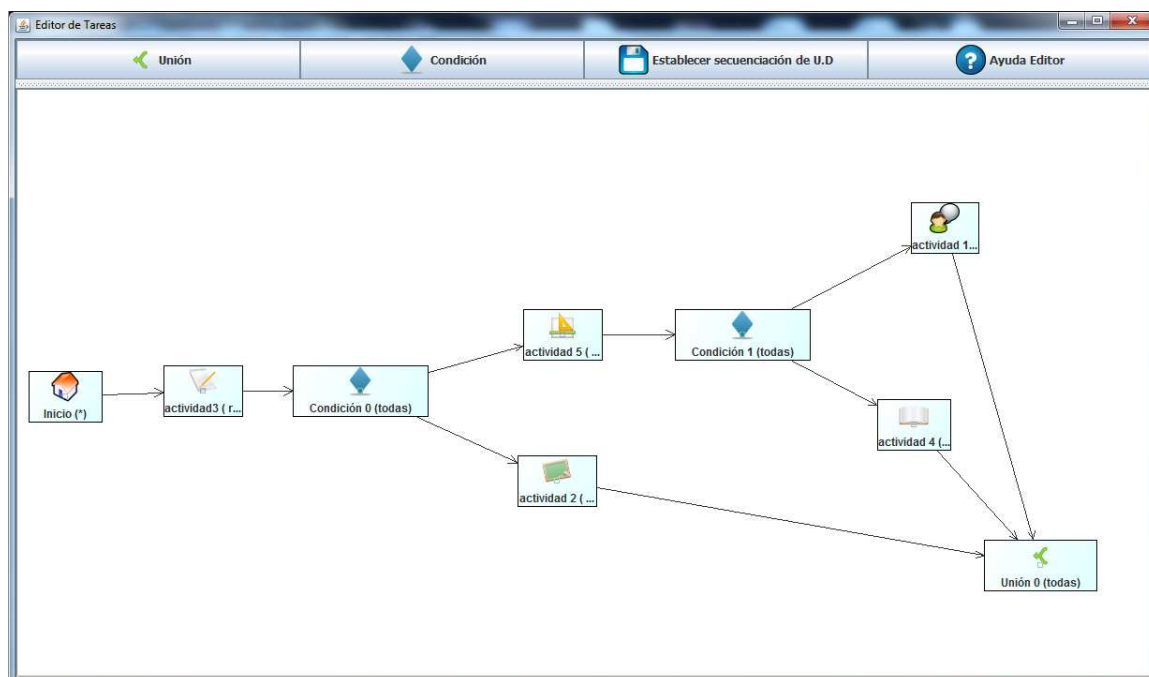


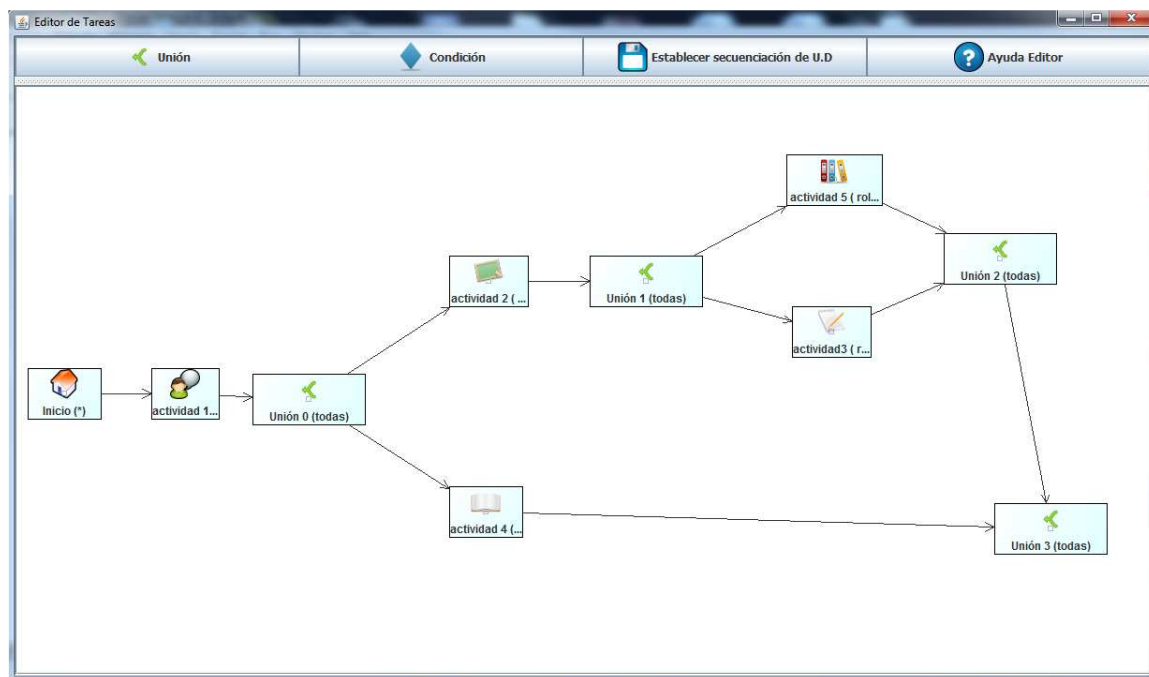
Figura 145 Editor gráfico. Secuenciación 9

- **Secuenciación 10**



*Figura 146 Editor gráfico. Secuenciación 10*

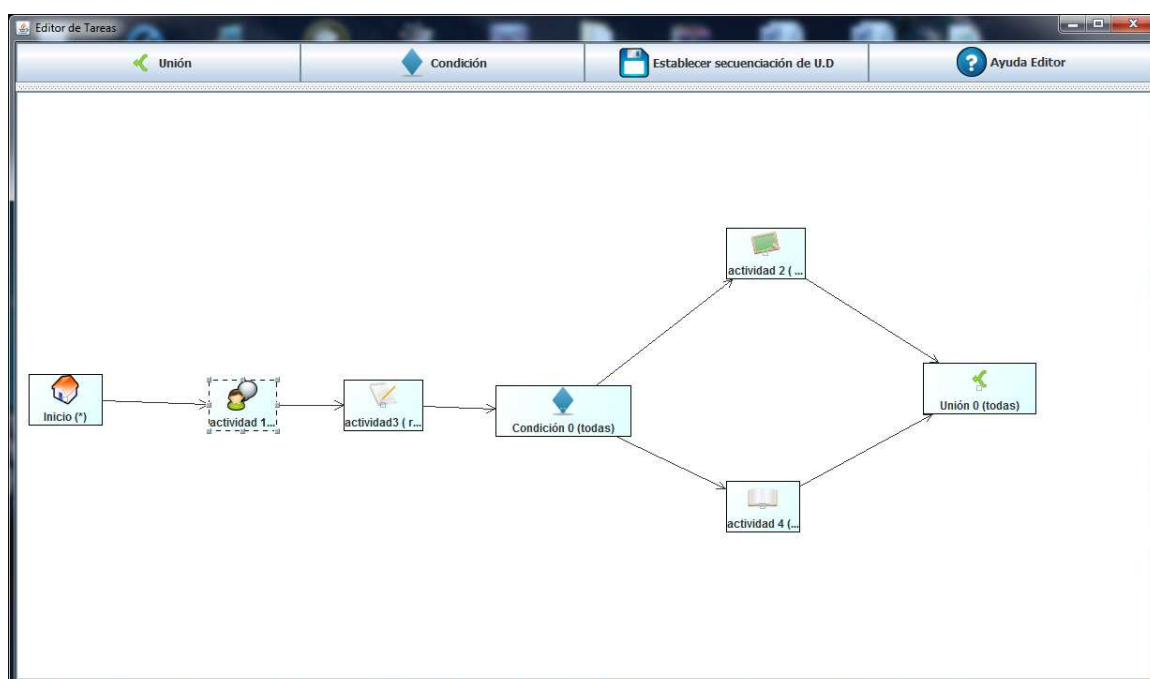
- **Secuenciación 11**



*Figura 147 Editor gráfico. Secuenciación 1*

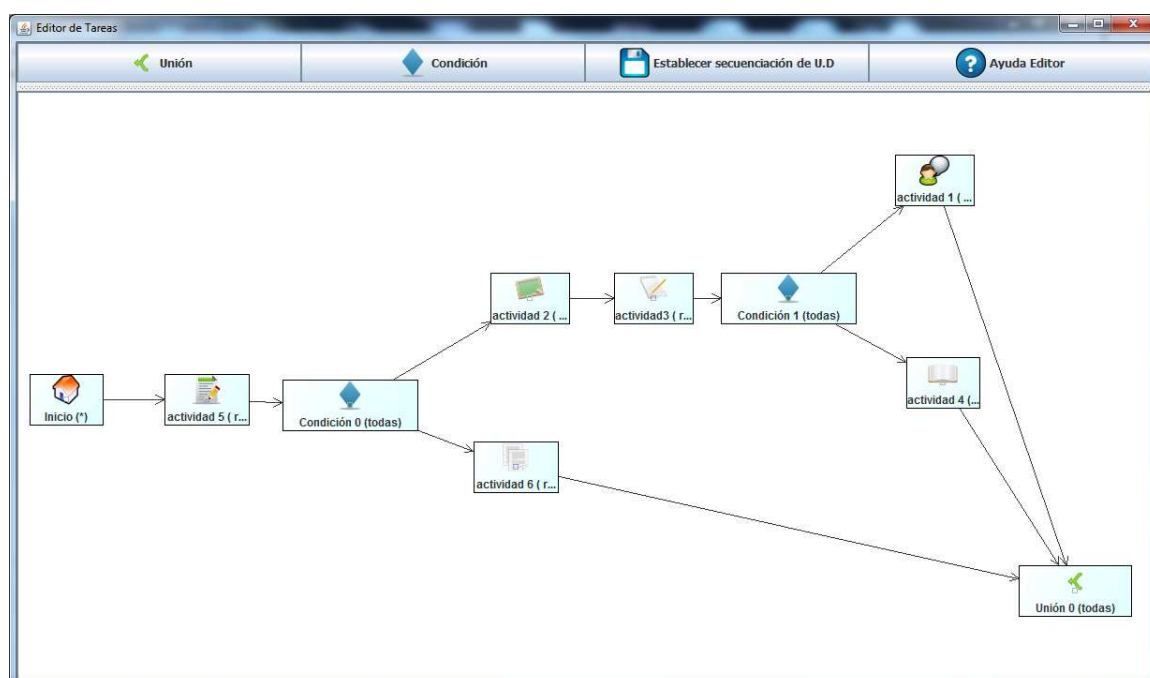


- **Secuenciación 12**



*Figura 148 Editor gráfico. Secuenciación 12*

- **Secuenciación 13**



*Figura 149 Editor gráfico. Secuenciación 13*

- **Secuenciación 14**

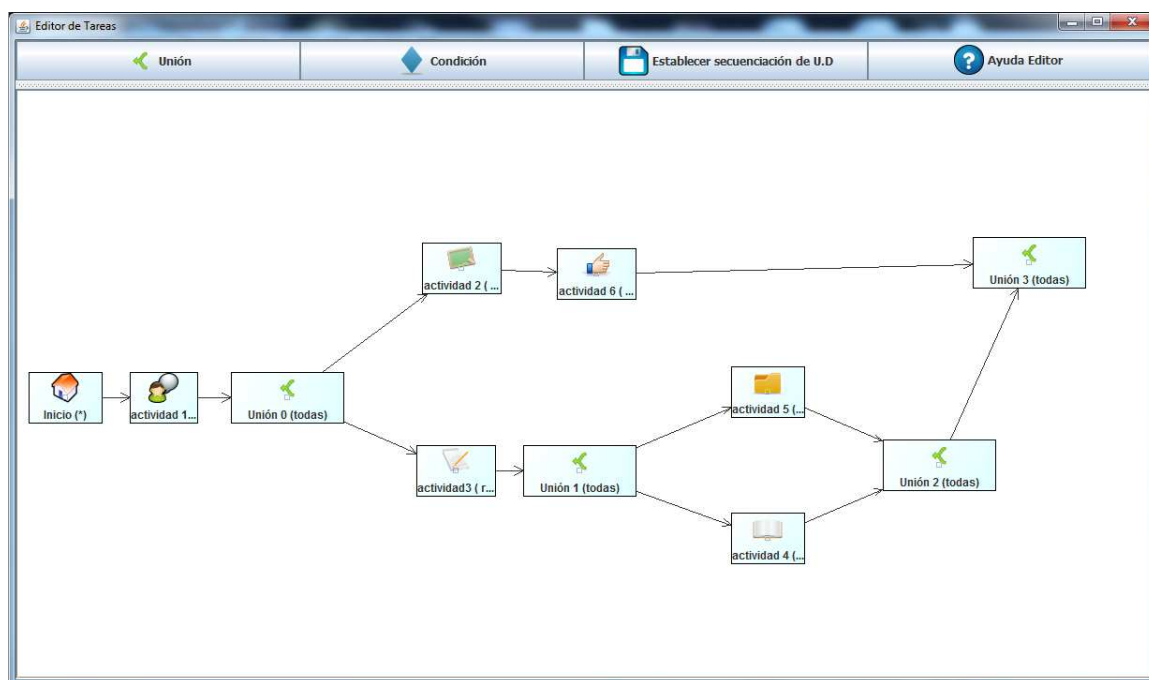


Figura 150 Editor gráfico. Secuenciación 14

- **Secuenciación 15**

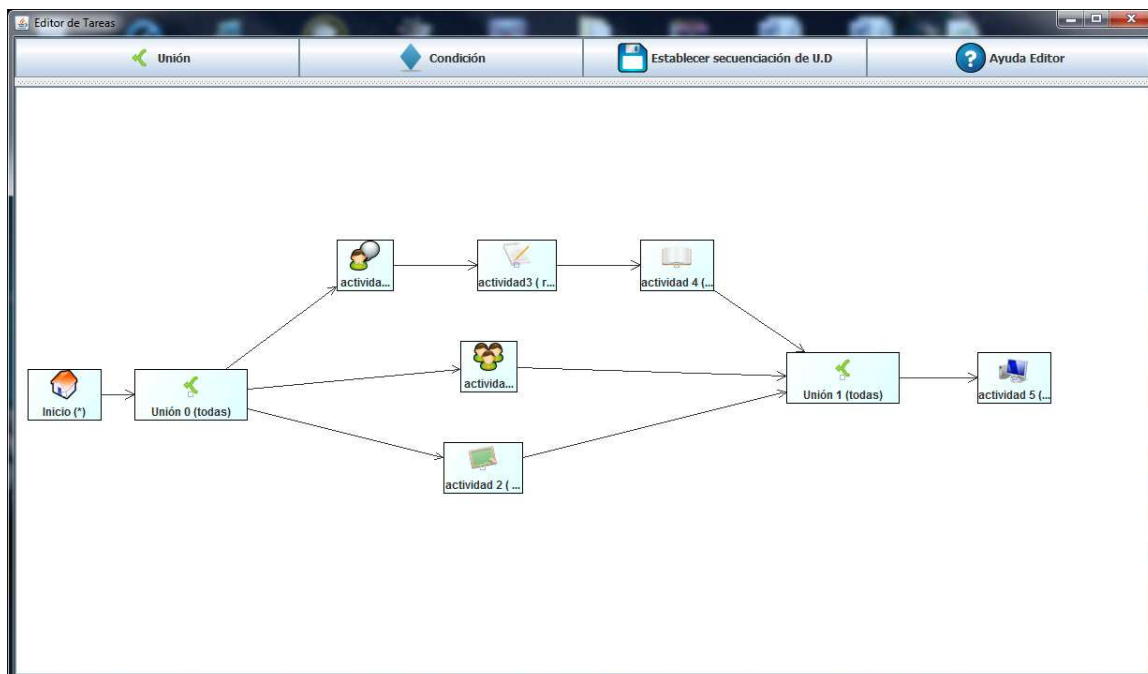
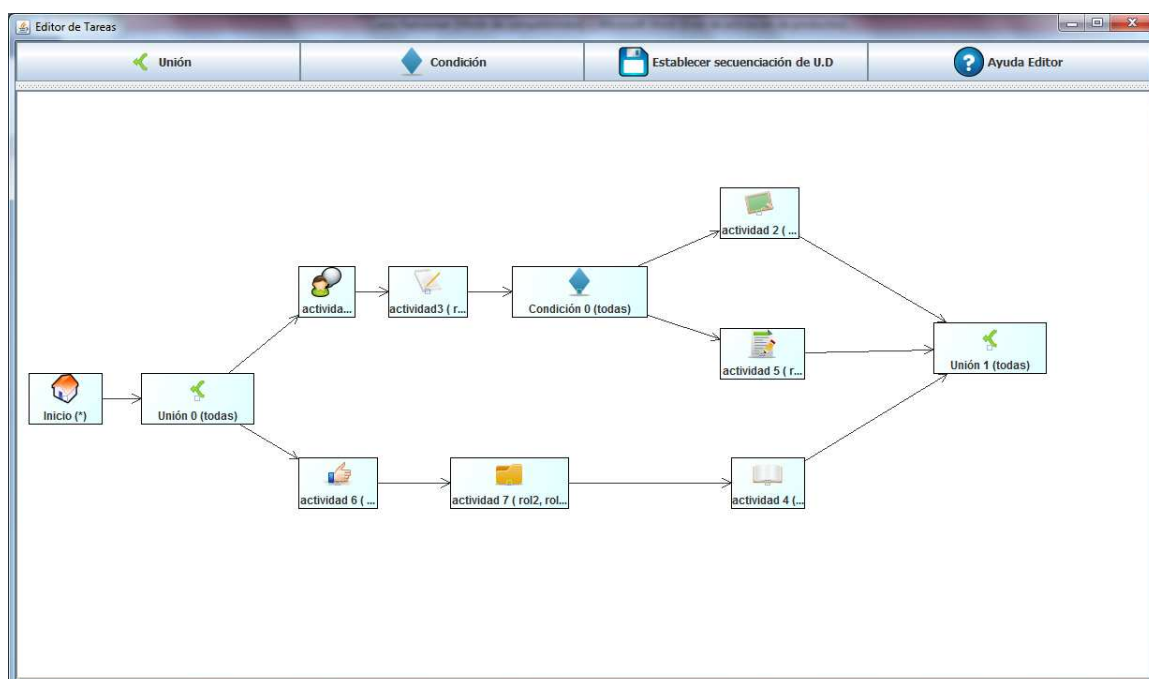


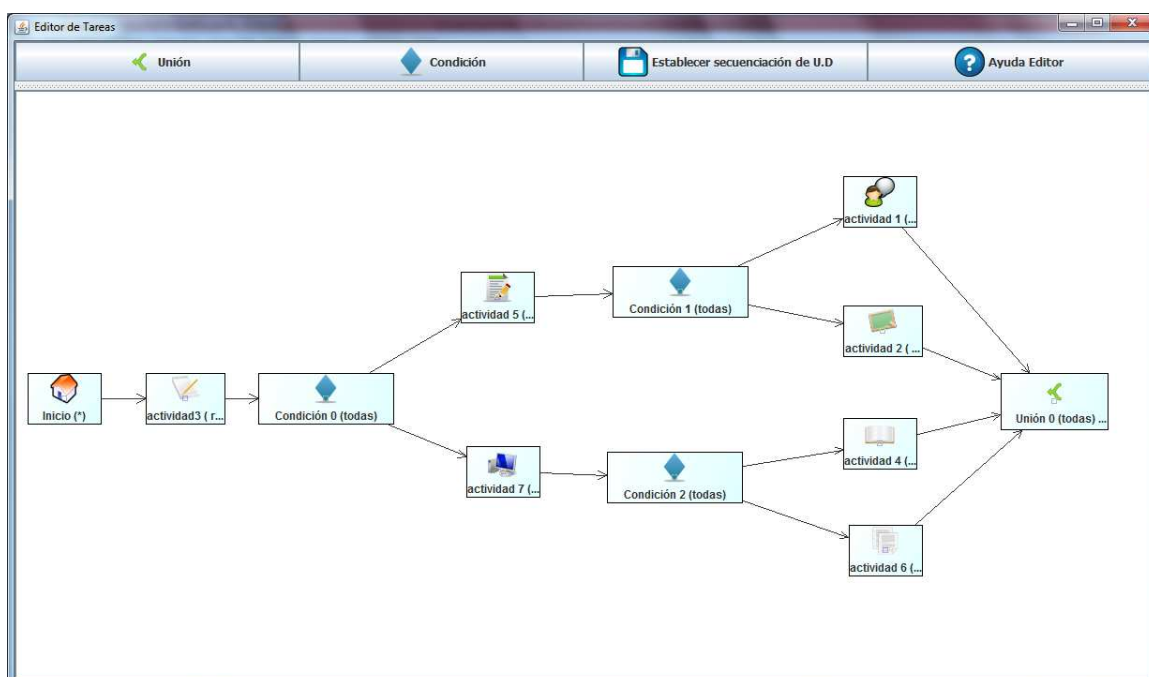
Figura 151 Editor gráfico. Secuenciación 15

- **Secuenciación 16**



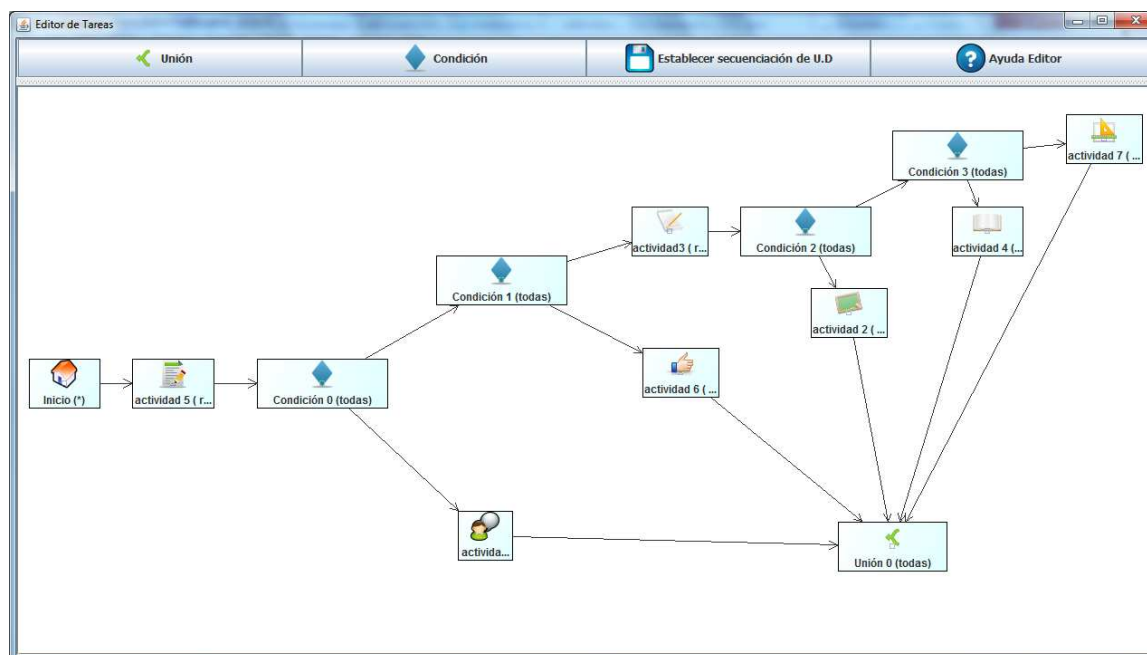
*Figura 152 Editor gráfico. Secuenciación 16*

- **Secuenciación 17**



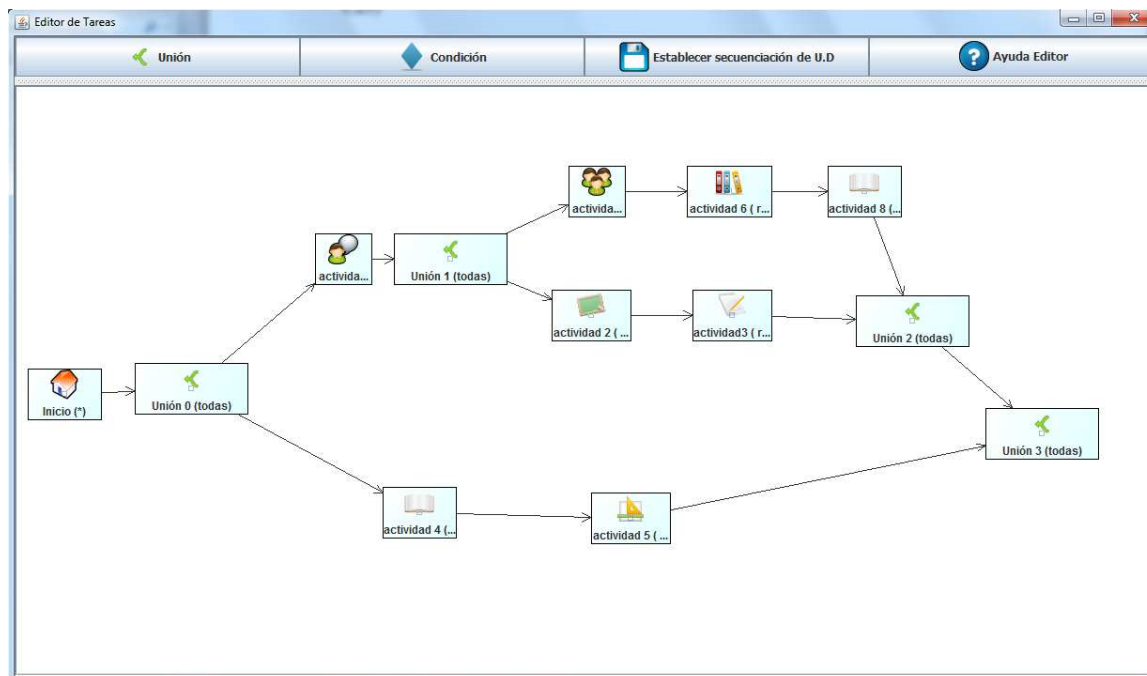
*Figura 153 Editor gráfico. Secuenciación 17*

- **Secuenciación 18**



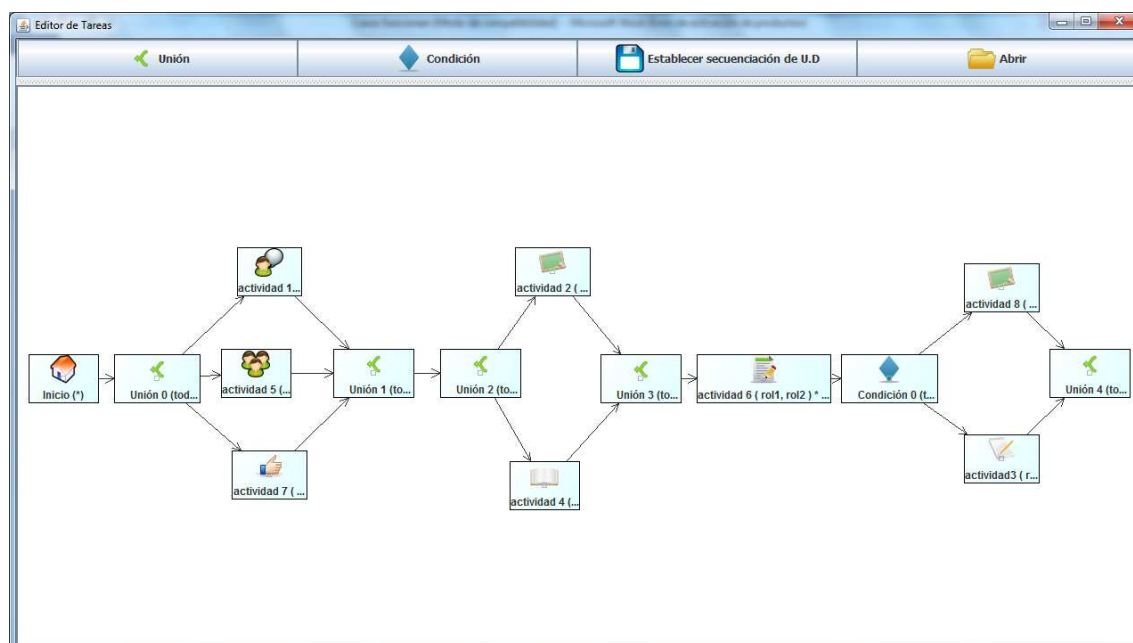
*Figura 154 Editor gráfico. Secuenciación 18*

- **Secuenciación 19**



*Figura 155 Editor gráfico. Secuenciación 19*

- **Secuenciación 20**



*Figura 156 Editor gráfico. Secuenciación 20*

# Glosario

IMS-LD	<i>IMS-Learning Design</i>
LMS	<i>Learning Management Systems</i>
LCMS	<i>Learning Content Management Systems</i>
CMS	<i>Content Management Systems</i>
XML	<i>eXtensible Markup Language</i>
TIC	<i>Tecnologías de la Información y la Comunicación</i>
ISO	<i>International Standards Organisation</i>
BSI	<i>British Standards Institute</i>
CEN	<i>Centre Europeande Normalisation</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IEEE LTSC	<i>Institute for Electrical and Electronic Engineers Learning Technology Standards Committee</i>
ADL	<i>Advanced Distributed Learning</i>
AICC	<i>Aviation Industry CBT Committee</i>
SCORM	<i>Shareable Content Object Reference Mode</i>
CAM	<i>Content Aggregation Model</i>
RTE	<i>Run-Time Environment</i>
SN	<i>Sequencing and Navigation</i>
CBT	<i>Computer-Based Training</i>
CMI	<i>Computer Managed Instruction</i>
W3C	<i>World Wide Web Consortium</i>
UoL	<i>Unit of Learning</i>
MS SQL Server	<i>Microsoft Structured Query Language Server</i>

HSQldb	<i>Hyper Structured Query Language Database</i>
UKOU	<i>UK Open University</i>
OUnL	<i>Open University Netherlands</i>
GNU	<i>General Public License</i>
GUI	<i>Graphical User Interface</i>
LDAP	<i>Lightweight Directory Access Protocol</i>
SAX	<i>Simple API for XML</i>
JDOM	<i>Java Document Object Model</i>
IDE	<i>Integrated Development Environment</i>
JDT	<i>Java Development Toolkit</i>
ECJ	<i>Eclipse Compiler for Java</i>

# Referencias

- [1] Desarrollo Curricular. Disponible [Internet]:  
<http://estatico.uned.ac.cr/paa/pdf/Materiales-autoev/17.pdf> [27 de Junio de 2013]
- [2] Baelo Álvarez, Roberto. *El e-learning, una respuesta educativa a las demandas de las sociedades del siglo XXI*. Universidad de León. Julio de 2009. Disponible [Internet]: <http://www.doredin.mec.es/documentos/00820103011025.pdf> [05 de Mayo de 2013]
- [3] García Peñalvo, Francisco José. *Estado actual de los sistemas e-learning*. Universidad de Salamanca. Disponible [Internet]:  
[http://campus.usal.es/~teoriaeducacion/rev\\_numero\\_06\\_2/n6\\_02\\_art\\_garcia\\_penalvo.htm](http://campus.usal.es/~teoriaeducacion/rev_numero_06_2/n6_02_art_garcia_penalvo.htm) [27 de Junio de 2013]
- [4] Artículo. Hornos Barranco, Miguel J. Montes Soldado, Rosana, Hurtado Torres, M. Visitación, Abad Grau, M. Mar. *E-Learning: nuevas tecnologías aplicadas a la formación en la empresa*. Facultad de Ciencias Económicas y Empresariales. Universidad de Granada. Disponible [Internet]:  
[http://lsi.ugr.es/rosana/investigacion/papers/elearn\\_efsi04.pdf](http://lsi.ugr.es/rosana/investigacion/papers/elearn_efsi04.pdf) [27 de Junio de 2013]
- [5] Accogli, Juan Ignacio. *Ventajas del e-learning*. Disponible [Internet]:  
[http://www.mental-gym.com/Docs/ARTICULO\\_80.pdf](http://www.mental-gym.com/Docs/ARTICULO_80.pdf) [05 de Mayo de 2013]
- [6] Sistemas e-learning. Desventajas del e-learning. Disponible [Internet]:  
<http://www.sistemaselearning.com/contenidos/Desventajas-del-e-learning.asp> [05 de Mayo de 2013]



- [7] ISO (International Organization for Standardization) Disponible [Internet]: <http://www.iso.org/iso/home.html> [05 de Mayo de 2013]
- [8] BSI (British Standards Institute) Disponible [Internet]: <http://www.bsigroup.es/> [05 de Mayo de 2013]
- [9] CEN (Centre Europeande Normalisation) Disponible [Internet]: <https://www.cen.eu/cen/pages/default.aspx> [05 de Mayo de 2013]
- [10] IEEE (Institute of Electrical and Electronics Engineers) Disponible [Internet]: <http://www.ieee.org/index.html> [05 de Mayo de 2013]
- [11] Fernández Manjón, Báltasar. *Especificaciones y estándares en e-learning*. Marzo de 2006. Disponible [Internet]: [http://reddigital.cnice.mec.es/6/Articulos/pdf/Articulos\\_2.pdf](http://reddigital.cnice.mec.es/6/Articulos/pdf/Articulos_2.pdf) [05 de Mayo de 2013]
- [12] IEEE Learning Technology Standards Committee (LTSC). Disponible [Internet]: <http://www.ieeeltsc.org:8080/Plone> [27 de Junio de 2013]
- [13] SCORM. Disponible [Internet]: <http://scorm.com/?gclid=COWPwa61hbgCFfLHtAodaA0A2g> [27 de Junio de 2013]
- [14] Ministerio de Educación y Ciencia. Disponible [Internet]: <http://ares.cnice.mec.es/informes/16/contenido/39.htm> [05 de Mayo de 2013]
- [15] IMS. Disponible [Internet]: <http://www.imsglobal.org/> [27 de Junio de 2013]
- [16] W3C. World Wide Web Consortium. Disponible [Internet]: <http://www.w3c.es/> [05 de Mayo de 2013]
- [17] Hilerá González, José Ramón; Hoya Marín, Rubén. *Estándares de e-learning. Guía de consulta*. Universidad de Alcalá. ISBN: 978-84-693-0263-7. Disponible [Internet]: <http://www.cc.uah.es/hilera/GuiaEstandares.pdf> [05 de Mayo de 2013]
- [18] IMS Learning Design Information Model. Disponible [Internet]: [http://www.imsglobal.org/learningdesign/ldv1p0/imsld\\_infov1p0.html](http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html) [27 de Junio de 2013]
- [19] Centro Aragonés de Tecnologías para la Educación. Disponible [Internet]: <http://www.catedu.es/webcatedu/index.php/descargas/herramientas-de-autor> [05 de Mayo de 2013]
- [20] CopperCore. Disponible [Internet]: <http://coppercore.sourceforge.net/> [05 de Mayo de 2013]
- [21] Java. Disponible [Internet]: <http://www.java.com/es/> [05 de Mayo de 2013]
- [22] Kai Toedter. JCalendar. Disponible [Internet]:

- <http://www.toedter.com/en/jcalendar/> [05 de Mayo de 2013]
- [23] JGraph. Disponible [Internet]: <http://www.jgraph.com/jgraph.html> [05 de Mayo de 2013]
- [24] JavaHelp. Disponible [Internet]: <http://javahelp.java.net/> [05 de Mayo de 2013]
- [25] XStream. Disponible [Internet]: <http://xstream.codehaus.org/index.html> [05 de Mayo de 2013]
- [26] JDom. Disponible [Internet]: <http://www.jdom.org/> [05 de Mayo de 2013]
- [27] JUnit. Disponible [Internet]: <https://github.com/junit-team/junit/wiki> [05 de Mayo de 2013]
- [28] Extensible Markup Language (XML). Disponible [Internet]: <http://www.w3.org/XML/> [27 de Junio de 2013]
- [29] Griffiths, David; Blat, Josep; García, Rocío; Sayago, Sergio. *La aportación de IMS Learning Design a la creación de recursos pedagógicos reutilizables*. Grupo de Tecnologías Interactivas. Universitat Pompeu Fabra. Disponible [Internet]: <http://revistas.um.es/red/article/view/24441/23781> [05 de Mayo de 2013]
- [30] Eclipse. Disponible [Internet]: <http://www.eclipse.org/org/> [05 de Mayo de 2013]
- [31] Bugzilla. Disponible [Internet]: <http://www.bugzilla.org/about/> [05 de Mayo de 2013]
- [32] Bugzilla. Laboratorio GAST UC3M. Disponible [Internet]: <https://bugzilla.gast.it.uc3m.es/> [05 de Mayo de 2013]
- [33] Lee, Nancy K. Interview JavaHelp. Disponible [Internet]: <http://www.stcsig.org/oi/hyperviews/archive/98Winter/java101.htm> [05 de Mayo de 2013]
- [34] Java Tutorials. *How to Use Tool Tips*. Disponible [Internet]: <http://docs.oracle.com/javase/tutorial/uiswing/components/tooltip.html> [05 de Mayo de 2013]
- [35] Icon Archive. Disponible [Internet]: <http://www.iconarchive.com/> [05 de Mayo de 2013]
- [36] Oracle. Disponible [Internet]: <http://www.oracle.com/technetwork/es/java/javase/downloads/index.html> [05 de Mayo de 2013]

